PATENT ABSTRACTS OF JAPAN

(11)Publication number : 10-063537
(43)Date of publication of application : 06.03.1998

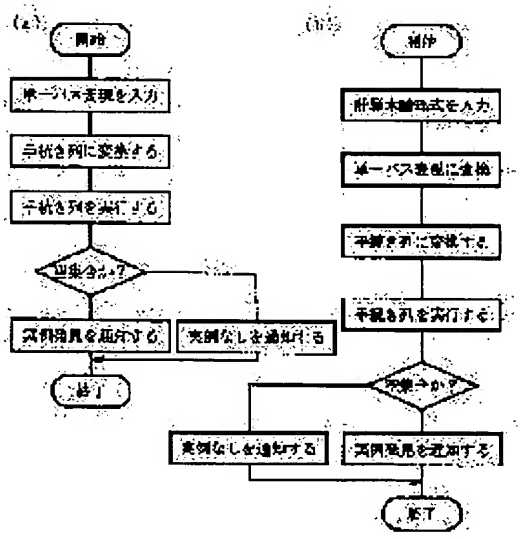
(51)Int.Cl. G06F 11/28
G06F 17/50

(21)Application number : 08-220005
(22)Date of filing : 21.08.1996

(71)Applicant : FUJITSU LTD
(72)Inventor : NAKADA TSUNEO
IWASHITA HIROAKI

(54) PROPERTY VERIFYING METHOD AND DEVICE THEREFOR

(57)Abstract:
PROBLEM TO BE SOLVED: To provide a property verifying method and device therefor capable of executing both of the reduction of a memory scale and that of a processing time in a symbol model examining method.
SOLUTION: In the property verifying method verifying whether a finite state machine expressing the operation of a synchronizing type ordering machine satisfies property expressing a function specification, a single path expression indicating a single state set string without branching or the state set string of constitution in which a limited loop is connected to the single state set string without branching is inputted. Then the method converts the single path expression into a procedure string obtained by successively combining prescribed procedures executable only by image arithmetic processing in the finite state machine, executes each procedure indicated by the procedure string, obtains a state set by way of the path of state transition corresponding to single path expression and gives information of the finding of an example or the absence of it depending on whether the obtained state set is a null set or not.



LEGAL STATUS

[Date of request for examination] 30.03.2000
[Date of sending the examiner's decision of rejection] 26.02.2002
[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]
[Date of final disposal for application]
[Patent number]

* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

CLAIMS

[Claim(s)]
[Claim 1] In a property verification method of verifying whether a finite state machine showing actuation of a synchronous system sequential machine filling a property showing functional specifications A single pass expression which shows a condition set train of a configuration of that an endless loop connected with one condition set train which does not have one condition set train or branching without branching as a property for verification is inputted. Change into a procedure train which combined a predetermined procedure which can perform said single pass expression only by image computation in said finite state machine one by one, and each procedure shown in said procedure train is performed. A property verification method characterized by asking for a condition set which goes via pass of a state transition corresponding to said single pass expression, and notifying example discovery or those without an example based on whether an obtained condition set is empty class.
[Claim 2] A property verification method characterized by inputting count tree logical expression, changing said count tree logical expression into a single pass expression according to predetermined transformation rule, and presenting transform processing to a procedure train as information showing a property for verification in a property verification method according to claim 1.
[Claim 3] In a property verification method of verifying whether a finite state machine showing actuation of a synchronous system sequential machine filling a property showing functional specifications Count tree logical expression is inputted as information showing a property for verification. Change into a procedure train which combined a predetermined procedure which can perform said count tree logical expression only by image computation in said finite state machine one by one according to predetermined transformation rule, and each procedure shown in said procedure train is performed. A property verification method characterized by asking for a condition set which goes via pass of a state transition corresponding to said single pass expression, and notifying example discovery or those without an example based on whether an obtained condition set is empty class.
[Claim 4] In a property verification method according to claim 2 or 3, it is based on predetermined conditions. Application of reverse image count of inputted count tree logical expression separates into a required subexpression and other count tree logical expression, and performs reverse image computation about said subexpression. A property verification method characterized by asking for a propositional-logic type showing a condition set train, compounding said propositional-logic type and count tree logical expression of said others, and presenting latter processing.
[Claim 5] Property verification equipment which verifies whether a finite state machine showing actuation of a synchronous system sequential machine characterized by providing the following is filling a property showing functional specifications A single pass expression input means to input a single pass expression which shows a condition set train of a configuration of that an endless loop connected with one condition set train which does not have one condition set train or branching without branching as a property for verification A procedure train conversion means to change said single pass expression into a procedure train which combined a predetermined procedure one by one A data-processing means to ask for a condition set which performs image computation in said finite state machine, and goes via pass

of a state transition corresponding to said single pass expression according to each procedure shown in said procedure train A judgment means to judge whether an obtained condition set is empty class
[Claim 6] It is property verification equipment carry out that a single pass expression input means is the configuration which it had in a logical-expression input means input count tree logical expression showing a property for verification, and an expression conversion means change said count tree logical expression into a single pass expression, and present processing of a procedure train conversion means with it according to predetermined transformation rule as the feature in property verification equipment according to claim 5.
[Claim 7] Property verification equipment which verifies whether a finite state machine showing actuation of a synchronous system sequential machine characterized by providing the following is filling a property showing functional specifications A logical expression input means to input count tree logical expression showing a property for verification A transform-processing means to change said count tree logical expression into a procedure train which combined a predetermined procedure one by one according to predetermined transformation rule A data-processing means to ask for a condition set which performs image computation in said finite state machine, and goes via pass of a state transition corresponding to said single pass expression according to each procedure shown in said procedure train A judgment means to judge whether an obtained condition set is empty class
[Claim 8] In property verification equipment according to claim 6 or 7 a logical expression input means A separation means to separate into a reception means to receive an input of count tree logical expression showing a property, and a subexpression and other count tree logical expression by which said count tree logical expression is needed for application of reverse image computation, A reverse image count means to ask for a propositional-logic type in which applying reverse image computation to said subexpression, and showing a condition set, Property verification equipment characterized by being the configuration equipped with a synthetic means to output as count tree logical expression with which said propositional-logic type and count tree logical expression of said others are compounded, and a property is expressed.

[Translation done.]

* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

DETAILED DESCRIPTION

[Detailed Description of the Invention]
[0001]
[The technical field to which invention belongs] It is related with the property verification method and equipment with which the model of the logical unit to which this invention was given with the finite state machine verifies whether the functional specifications of a logical unit are filled. A finite state machine is defined as a finite automaton with an output, and is expressed with Set Q and the input alphabet sigma of a condition, the output alphabet phi, the transition relational expression delta, the output relational expression lambda, and 6 **** of the initial-state set I like a formula (1).
M= (Q, sigma, phi, delta, lambda, I) ... (1)
Since non-deterministic transition is expressed, the transition relational expression delta mentioned above is a function which determines the following condition by taking 1 as a value, if it may change in degree condition when the input is given in the present condition, and taking 0 as a value, if there is nothing, when an actual condition object, degree condition, and an input are given. Here, since all synchronous sequential circuits can be modeled with a finite state machine in principle, in case a logical unit is designed, the method of using a finite state machine as the specification is used widely. For example, in the circuit design by logic synthesis, the method of changing into a finite state machine the specification of equipment expressed with design-description language, and a flip-flop and a register realizing each condition, and realizing transition relational expression and output relational expression in a combinational circuit is common. Thus, the verification activity of the model of the designed logical unit consists of various phases from the verification activity about abstract specification to the verification activity about actuation of an actual circuit. these phases -- inside, since verification of functional specifications can expect the error detection in the earliest phase of layout of a logical unit, the quick and positive verification method is needed.
[0002]
[Description of the Prior Art] As the technique of verifying the functional specifications of a logical unit, it roughly divides and the logic simulation technique and the formal verification technique are mentioned. It is the technique of verifying the justification of the model of a logical unit by investigating whether the logic simulation technique is in agreement with the output from which a suitable input is applied to the model of a logical unit, actuation of a logical unit is simulated, and the obtained output is obtained from the original finite state machine.
[0003] Therefore, by the logic simulation technique, the activity which creates input data in consideration of various conditions is indispensable, and cannot verify about the condition which was not taken into consideration in creation of this input data. On the other hand, the formal verification technique verifies mathematically whether the property which the finite state machine whose model of a logical unit is origin is filling is realized.
[0004] in addition, formal verification -- being related -- for example, "formal verification technique based on logical function processing" Hiraishi, Hamaguchi, information processing, and Vol.35 (8):pp710- please refer to reference, such as 718 and 1994 (reference 1 is called hereafter). The symbolic

model inspection technique which expresses a property as the formal verification technique in the count tree logic which is a kind of tense logic, With omega-automaton which extended acceptance conditions that it should correspond to the input train of infinity A property The language inclusion inspection technique to express (R.) [Hojati, R.KBrayton,] [and R.P.Kurshan."BDD-based debugging of designs using languagecontainment] and fair CTL." In Proceedings of the Conference on ComputerAided Verification and 1993 are proposed.

[0005] It is the technique of verifying whether the model of a logical unit satisfies specification by the symbolic model inspection technique's expressing the logical unit model of Kripke structure using logical function, and inspecting whether the condition set which is not the empty with which are satisfied of the specification expressed with count tree logic exists among the formal verification technique. Here, the Kripke structure K is the finite set S of a condition, and the transition relation R between conditions and the set S0 of an initial-state point. It is a kind of the non-deterministic finite automaton expressed with each condition like a formula (2) using the set L of the primitive proposition used as truth.

[0006]

K= (S, R, S0, L) ... (2)

moreover, count tree logic -- a kind of tense logic -- it is -- the usual logical operator -- in addition, the tense operator F with which it expresses "when it is" with the operator E showing the operator A showing all **, and existence -- "-- always -- " -- it is expressed using the tense operator U showing the period until ["until"] the tense operator X showing the tense operator G and the "degree" to express.

[0007] For example, the tense logic AGa shows the purport in which logical expression a is materialized in all the condition sets that can reach from an initial state. In this case, in the model of a logical unit, all the routes that can change from an initial state are followed, and all those routes should just investigate whether it arrives at the condition that logical expression a is materialized.

[0008] That is, the verification activity in the symbolic model inspection technique follows the state transition of Kripke structure, in each condition, it is the activity which checks whether the count tree logical expression showing specification is materialized, and this activity can result in the set operation which asks for the minimum fix point or the maximum fix point on the model which used the formula of count tree logic. Such a set operation is realized by combining the image count Image ({q}) which asks for the condition set which can reach by one transition from a certain condition set {q}, and reverse image count Image-1 ({q}) which ask for the condition set which can reach a certain condition set {q} by 1 time of the state transition.

[0009] For example, as shown in drawing 11 , it is nine condition q0 -q8. In the example of a finite state machine expressed with the state transition of a between, the example of the result of having performed image count and reverse image count is shown in a formula (6) from a formula (3).

Image({q0}) = {q0, q1, q2, q3} ... (3)

Image({q0, q8}) = {q2, q3, q5, q6, q8} ... (4)

Image-1({q0}) = {q0, q1} ... (5)

Image-1({q5}) = {q1, q2, q3, q4} ... (6)

the finite state machine shown in this drawing 11 -- setting -- for example, condition q7 the time of verifying the tense logic AF p using the shown logical expression p -- initial state q0 from -- all the routes that can change image count from a repeat and an initial state one by one -- condition q7 What is necessary is just to investigate whether it reaches or not. the case where the tense logic EF p is verified on the other hand -- condition q7 from -- reverse image count -- repeating -- initial state q0 What is necessary is just to investigate whether there is any reaching route.

[0010] By the way, in the actual symbolic model inspection technique, when the technique of transposing a set operation to logical function processing is common (reference 1 reference) and expresses logical function in a bisection decision graph, the increase in efficiency of logical function processing is attained (R. E.Bryant."Graph based algorithm for boolean function manipulation." IEEE Trans.Comput., C-35(8):pp 677-691, 1986.).

[0011] Moreover, the technique of aiming at the cutback of the magnitude of a bisection decision graph

which expresses degree condition as a function of the present condition and an input (the input of a dimension and new input), and expresses state-transition relation is also proposed by applying and controlling a new input into a state-transition-related non-deterministic portion.
[0012]
[Problem(s) to be Solved by the Invention] As mentioned above, when a bisection decision graph expresses non-deterministic transition relation, as shown in the column of "having no function-izing" of drawing 12 , the image count and reverse image count which are used as a verification procedure can be processed by the time amount of the almost same order.
[0013] However, in this case, it depends on an input variable for the number of nodes which shows the magnitude of a bisection decision graph, and in being the worst, the number of nodes increases exponentially according to buildup of an input variable. For this reason, since memory space huge since a bisection decision graph is stored was needed when it applies to the model of a logical unit with the large number of conditions also in the model for an experiment, if it remained as it is, having treated with a realistic computing system was almost impossible.
[0014] It sets to drawing 12 and is 300MB. It is Sign space to the column which corresponds about the model (for example, vpp, pipe-1) it became impossible since the above memory space was needed verifying. It was given and shown. While the magnitude of a bisection decision graph can be suppressed on the other hand in the magnitude which can be processed with a realistic computing system as shown in a column "with function-ized" when state-transition relation is expressed functionally and the cutback of the magnitude of a bisection decision graph is aimed at, the processing time which reverse image count takes increases remarkably. [of drawing 12] Since [the] the reverse image count is used abundantly in the verification activity in the symbolic model inspection technique rather than image count, the time amount which the whole verification activity takes will increase substantially, and it will become impossible to treat by the throughput of a realistic computing system on the other hand according to buildup of the time amount which reverse image count takes.
[0015] For this reason, in having used the conventional technique as it was, the magnitude or the processing time needed of the memory needed exploded, and verifying the logical unit of the magnitude equivalent to an actual processor etc. was hardly completed.
[0016] By the way, in the conventional technique mentioned above, it is all premised on enabling a verification activity corresponding to all the properties that can be expressed by count tree logic. However, it turns out experientially that it is [of the property which can express a property to verify in a logical unit by count tree logic] a part very much.
[0017] For example, tense logical expression "EF p" shows "the purport which can reach the condition set Qp of which logical expression p consists from an initial state", shows the prohibition condition that a logical unit must not fall with logical expression p, and when verifying the property of a purport with which the model of a logical unit does not lapse into a prohibition condition, it is used frequently. Moreover, tense logical expression "AG (p->AF q)" is frequently used, when verifying the property of the purport which a response surely generates, if the purport "which passes along the condition set in which logical expression q surely consists of the subset which can reach from an initial state among the condition sets of which logical expression p consists even if it passes along what kind of pass" is shown, for example, a request occurs.
[0018] The example of the tense logical expression "EF p" mentioned above is an initial state to the condition set Qp. It becomes the condition set train in which it results, and on the other hand, the "AG (p->AF q)" counter example serves as a condition set train which always results in the loop of which logical expression **q consists, after passing along the condition set in which logical expression p consists of an initial state. [of tense logical expression] Each of these condition set trains has composition which the condition set train used as an endless loop connected with the tail of one condition set train without branching, or such one condition set train. The property which the configuration of such a condition set train is called single pass, and can express an example or a counter example by the condition set train of a single pass configuration hereafter is called the property of a single pass expression.

[0019] By the way, according to presumption of artificers, it is thought that the property of a single pass expression which was mentioned above occupies 95% from 90% of the property set as the object of a verification activity in the model of a logical unit. Therefore, if the efficiency of the verification about the property of a single pass expression can be increased, it is possible to increase substantially the efficiency of the whole verification activity of a logical unit model.
[0020] This invention aims at offering the property verification method and equipment compatible in the cutback of memory magnitude, and compaction of the processing time in symbolic model detection method.
[0021]
[Means for Solving the Problem] Drawing 1 (a) and (b) It is drawing showing a principle of a property verification method of claim 1 and claim 2.
[0022] In a property verification method of verifying whether invention of claim 1 filling a property with which a finite state machine showing actuation of a synchronous system sequential machine expresses functional specifications A single pass expression which shows a condition set train of a configuration of that an endless loop connected with one condition set train which does not have one condition set train or branching without branching as a property for verification is inputted. Change into a procedure train which combined a predetermined procedure which can perform a single pass expression only by image computation in a finite state machine one by one, and each procedure shown in a procedure train is performed. It asks for a condition set which goes via pass of a state transition corresponding to a single pass expression, and is characterized by notifying example discovery or those without an example based on whether an obtained condition set is empty class.
[0023] Invention of claim 1 can ask for a condition set which fills a property by performing image computation according to a procedure train which changed an inputted single pass expression and was acquired, and can verify a property based on whether this condition set is empty class. That is, if a single pass expression of a property for verification is possible, since the verification activity is realizable only using image computation, in symbolic model detection method, it will become possible to be compatible in a cutback of memory magnitude and compaction of the processing time by having applied degradation of a bisection decision graph by state-transition-related function-izing.
[0024] In a property verification method according to claim 1, as information showing a property for verification, count tree logical expression is inputted, and invention of claim 2 changes count tree logical expression into a single pass expression according to predetermined transformation rule, and is characterized by presenting transform processing to a procedure train. Since invention of claim 2 can change count tree logical expression into a single pass expression and can present transform processing to a procedure train with it, it becomes possible [using for verification of a property expressed with count tree logical expression].
[0025] Drawing 2 (a) and (b) It is drawing showing a principle of a property verification method of claim 3 and claim 4. In a property verification method of verifying whether invention of claim 3 filling a property with which a finite state machine showing actuation of a synchronous system sequential machine expresses functional specifications Count tree logical expression is inputted as information showing a property for verification. Change into a procedure train which combined a predetermined procedure which can perform count tree logical expression only by image computation in a finite state machine one by one according to predetermined transformation rule, and each procedure shown in a procedure train is performed. It asks for a condition set which goes via pass of a state transition corresponding to a single pass expression, and is characterized by notifying example discovery or those without an example based on whether an obtained condition set is empty class.
[0026] By considering a process in which count tree logical expression is directly changed into a procedure train, invention of claim 3 can reduce procedures required for verification of a property expressed with computer logical expression, and can increase further the efficiency of a property verification activity in which a single pass expression is possible. Invention of claim 4 is set to a property verification method according to claim 2 or 3. Based on predetermined conditions, application of reverse image count of inputted count tree logical expression separates into a required subexpression

and other count tree logical expression, and performs reverse image computation about a subexpression. It is characterized by asking for a propositional-logic type showing a condition set train, compounding a propositional-logic type and other count tree logical expression, and presenting latter processing.
[0027] Invention of claim 4 can cancel a limit about a property for verification by applying reverse image computation only about some subexpressions of count tree logical expression, compounding a propositional-logic type obtained as the result to the original computer logical expression, and presenting verification processing of a property. It becomes possible to make applicable to verification all properties that can be expressed with count tree logical expression by this.
[0028] Drawing 3 is drawing showing a principle of property verification equipment of claim 5 and claim 6. In property verification equipment which verifies whether invention of claim 5 is filling a property with which a finite state machine showing actuation of a synchronous system sequential machine expresses functional specifications A single pass expression input means 111 to input a single pass expression which shows a condition set train of a configuration of that an endless loop connected with one condition set train which does not have one condition set train or branching without branching as a property for verification, A procedure train conversion means 112 to change a single pass expression into a procedure train which combined a predetermined procedure one by one, According to each procedure shown in a procedure train, image computation in a finite state machine is performed. It is characterized by having a data-processing means 113 to ask for a condition set which goes via pass of a state transition corresponding to a single pass expression, and a judgment means 114 to judge whether an obtained condition set is empty class.
[0029] The property verification method of having stated at claim 1 can apply, and the property shown with a single pass expression can verify by invention of claim 5 changing into a procedure train the single pass expression which received through the single pass expression input means 111 with a procedure train conversion means 112, and offering the condition set from which the data-processing means 113 operated and it was obtained to processing of a judgment means 114 according to this procedure train.
[0030] Since the verification activity is realizable by this about a property in which a single pass expression is possible only using image computation, in symbolic model detection method, it becomes possible to be compatible in a cutback of memory magnitude and compaction of the processing time by having applied degradation of a bisection decision graph by state-transition-related function-izing.
[0031] Invention of claim 6 carries out that a single pass expression input means 111 is the configuration of having had a logical-expression input means 121 input count tree logical expression showing a property for verification, and an expression conversion means 122 change count tree logical expression into a single pass expression, and present processing of the procedure train conversion means 112 with it according to predetermined transformation rule as the feature in property verification equipment according to claim 5.
[0032] According to an input of computer logical expression by the logical expression input means 121, invention of claim 6 can verify a property expressed with count tree logical expression, when the expression conversion means 122 operates. Drawing 4 is drawing showing a principle of property verification equipment of claim 7 and claim 8. In property verification equipment which verifies whether invention of claim 7 is filling a property with which a finite state machine showing actuation of a synchronous system sequential machine expresses functional specifications A logical expression input means 121 to input count tree logical expression showing a property for verification, A transform-processing means 123 to change count tree logical expression into a procedure train which combined a predetermined procedure one by one according to predetermined transformation rule, According to each procedure shown in a procedure train, image computation in a finite state machine is performed. It is characterized by having a data-processing means 113 to ask for a condition set which goes via pass of a state transition corresponding to a single pass expression, and a judgment means 114 to judge whether an obtained condition set is empty class.
[0033] Invention of claim 7 becomes possible [doing still more nearly promptly a verification activity of a property expressed with count tree logical expression], when the transform-processing means 123

changes directly into a procedure train count tree logical expression received through the logical expression input means 121 and eliminates the need of going via a single pass expression. Invention of claim 8 is set to property verification equipment according to claim 6 or 7. The logical expression input means 121 A reception means 124 to receive an input of count tree logical expression showing a property, A separation means 125 to separate into a subexpression and other count tree logical expression by which count tree logical expression is needed for application of reverse image computation, Reverse image computation is applied to a subexpression, a reverse image count means 126 to ask for a propositional-logic type showing a condition set, and a propositional-logic type and other count tree logical expression are compounded, and it is characterized by being the configuration equipped with a synthetic means 127 to output as count tree logical expression showing a property.
[0034] It is convertible to the count tree logical expression which can single pass express the count tree logical expression containing a subexpression which needs reverse image count by the separation means 125 separating a part of count tree logical expression received through the reception means 124, and invention of claim 8 presenting processing of the reverse image count means 126 with it, and compounding the propositional-logic type obtained by reverse image count by the synthetic means 127 to the original count tree logical expression.
[0035] This cancels a limit about a property used as an object for verification, and since a verification activity can be done according to an input of count tree logical expression showing all properties, property verification equipment excellent in versatility is realizable.
[0036]
[Embodiment of the Invention] Hereafter, the operation gestalt of this invention is explained to details based on a drawing.
[0037] Drawing 5 is drawing showing the operation gestalt of the property verification equipment of claim 6. In drawing 5 , property verification equipment changes into a single pass expression the count tree logical expression received through the input reception section 211 equivalent to the logical expression input means 121 by the single pass expression converter 212. Furthermore, it has the composition of verifying the inputted property, by presenting processing of the set operation section 214, after changing into the procedure train later mentioned by the procedure train converter 213, and judging whether the obtained condition set is empty by the judgment processing section 215.
[0038] In the input reception section 211 shown in drawing 5 , the operator distinction section 221 has composition which sends out this count tree logical expression to the initial-state adjunct 223 through the direct or negation processing section 222 according to the operator of the head of the received count tree logical expression. Here, the operator distinction section 221 sends out the whole count tree logical expression to the negative processing section 222, when a top operator is a operator "A" which shows all **, and on the other hand, when it is the operator "E" which shows existence, it should just send it out to the initial-state adjunct 223 as it is.
[0039] Moreover, in drawing 5 , the negative processing section 222 denies the whole count tree logical expression received from the operator distinction section 221, and has composition which transforms a required formula according to a logical operation regulation, arranges count tree logical expression, and sends out the obtained count tree logical expression to the initial-state adjunct 223 further. By adding the propositional-logic type pi corresponding to an initial state to the head of the received count tree logical expression, this initial-state adjunct 223 makes the initiation event of a verification activity an initial state, and has composition which sends out this count tree logical expression to the single pass expression converter 212.
[0040] Moreover, in the single pass expression converter 212 shown in drawing 5 , the formula deformation processing section 224 transforms this count tree logical expression according to the transformation rule held in count tree logical expression from the input reception section 211 mentioned above at reception and the transformation-rule storing section 225, and has composition changed into a single pass expression. What is necessary is here, just to store in the transformation-rule storing section 225 the transformation rule shown in a formula (10) from a formula (7) using the propositional-logic types p and q showing a condition set, and the subexpression f of count tree logical expression

mentioned above as procedure R which changes count tree logical expression into a single pass expression.
[0041]
R(p**EX f) =pR (f) ... (7)
R(p**EF f) = {R (p**f), pTrue* R (f)} ... (8)
R(p**EG q)=(p**f)qomega ... (9)
R(p**E (q U f)) = {R (p**f), pq* R (f)} ... (10)
From the formula (7), the repeat of the finite time of the propositional-logic type p and an infinity time is bundled with a characteristic sign "*" and a characteristic sign "omega", a division is bundled with a brace an example and a case, and the formula (10) is shown.
[0042] Moreover, the formula deformation processing section 224 should just perform transform processing to a single pass expression by detecting the part which suits the transformation rule mentioned above from the received count tree logical expression, and applying the corresponding transformation rule, respectively.
[0043] Here, a single pass expression is the condition set train which the condition set train used as an endless loop connected with one condition set train without branching, or one condition set train without branching, and the set S of the single pass expression s is defined as follows using the set P of the propositional-logic type showing the condition set of arbitration. The condition set p which is p**P is a single pass expression.
[0044] The connection [train / s / which is the condition set p and s**S which are p**P / condition set] ps is a single pass expression. p omega of repeats of the infinity time of the condition set p which is p**P It is a single pass expression. Repeat p* of the finite time of the condition set p which is p**P Connection p* s with the condition set train s which is s**S is a single pass expression.
[0045] Therefore, what is necessary is for the formula deformation processing section 224 to transform the formula according to the transformation rule mentioned above, and just to search for the condition set train expressed with the single pass expression defined above. For example, when the count tree logical expression "AG (p->AFq)" showing the property which should be verified through the input reception section 211 is inputted Since a top operator is a operator "A" which shows all **, the negative processing section 222 operates according to the directions from the operator distinction section 221, and it is drawing 6 (a). So that it may be shown By arranging, once denying the inputted count tree logical expression, the count tree logical expression which does not contain a operator "A" is obtained. Furthermore, propositional-logic type pi corresponding to [to the head of this count tree logical expression] an initial state by the initial-state adjunct 223 It is added and processing of the formula deformation processing section 224 is presented with the obtained count tree logical expression.
[0046] According to the input of this count tree logical expression, the formula deformation processing section 224 operates and it is drawing 6 (b). It is drawing 6 (b) by setting and arranging to the subexpression attaching and showing a slash, respectively with the application of the formula (8) and formula (9) which were mentioned above one by one. The shown single pass expression is obtained. Thus, when each part of a single pass expression converter 212 operates, the function of the expression conversion means 122 which stated by claim 6 realizes, the count tree logical expression which received through the input reception section 211 can change into a single pass expression, the function of the single pass expression input means 111 which stated by claim 5 can realize as a whole, and it can offer to the processing of a procedure train converter 213 of corresponding to a procedure train conversion means 112.
[0047] In the procedure train converter 213 shown in drawing 5 , the formula deformation processing section 226 has composition changed into the verification procedure train which describes a single pass expression below according to the transformation rule held at the transformation-rule storing section 227. Here, a fundamental verification procedure train is the operator Img which is constituted by only image count and shows the propositional-logic types p and q and image count. And the operator gfp with which the maximum fix point and the minimum fix point are expressed, respectively and lfp It uses and is expressed like a formula (11) to a formula (13).

[0048]
FindTrans(p) =Img(p) ... (11) FindTrail(p, q) =lfp Z. [p**Img (q**Z)] ... (12) FindLoop (p) =gfp Z. [p**Img (Z)] ... verification procedure FindTrans (p) shown in (13) types (11) It is the image count itself and is just drawing 7 (a). It is the procedure which returns the condition set which can reach by 1 time of the state transition from the condition set expressed with the propositional-logic type p so that it may be shown.
[0049] Moreover, verification procedure FindTrail (p, q) shown in the formula (12) is drawing 7 (b). It is the procedure which returns the condition set (a slash is attached and shown in drawing 7 (b)) which can reach from the condition set expressed with the propositional-logic type p via the condition set of which the propositional-logic type q consists so that it may be shown. Moreover, verification procedure FindLoop shown in the formula (13) (p) Drawing 7 (c) It is the procedure which returns the sum-set (a slash is attached and shown in drawing 7 (c)) of the subset which constitutes the loop of which it is contained in the condition set expressed with the propositional-logic type p, and the propositional-logic type p consists, and the subset which can reach from the loop so that it may be shown.
[0050] Moreover, the transformation-rule storing section 227 stores the transformation rule S expressed like a formula (14) - a formula (18) using the single pass expression s and the propositional-logic type p.

S (p) =p ... (14) S (sp) =FindTrans(S (s)) **p ... (15) S(sp*) =FindTrail (it FindTrans(es) (S (s))) p) ... (16) S(spomega) =FindLoop (FindTrail(S (s), p) **p) ... (17) S({s**p, sp*}) =FindTrail (S (s), p) ... (18), therefore the formula deformation processing section 226 should just change a single pass expression into a verification procedure train by detecting the portion which suits these transformation rules from the received single pass expression, and applying the corresponding transformation rule one by one.
[0051] For example, drawing 6 (b) When the formula deformation processing section 226 applies a formula (17) and a formula (18) one by one and arranges a formula further according to the input of the shown single pass expression, as it is shown in drawing 6 (c), it can change into a procedure train. Thus, what is necessary is whether a condition set is empty class and for the set operation section 214 to perform the image count and fix-point count which were shown in the formula (13) from the formula (11) mentioned above, to ask for the condition set expressed with a single pass expression, to input it into the judgment processing section 215, and just to judge according to the input of the acquired procedure train.
[0052] Thus, the property verification equipment which applied the method of claim 1 is realizable. When state-transition relation is expressed functionally and the magnitude of a binary tree decision graph is reduced since these procedure trains can be performed only using image count as mentioned above, set operation processing can be performed using the throughput of a realistic computer system, and it can verify whether the property which can be expressed with a single pass expression by the model of a logical unit is realized.
[0053] Since it can be compatible by this in the control of the amount of memory and the compaction of the processing time which pose a problem about the property in which a single pass expression is possible in case the symbolic model inspection technique is applied, it becomes possible to apply the symbolic model inspection technique to property verification of a logical unit with realistic magnitude.
[0054] Here, as mentioned above, a single pass expression is possible for most properties for which verification is needed in an actual logical unit, and it can increase the efficiency of the whole verification activity substantially by attaining the increase in efficiency of a verification activity with the application of the technique of this invention to the verification activity of these properties. Furthermore, it is also possible to change count tree logical expression into the procedure train directly mentioned above.
[0055] The operation gestalt of the property verification equipment of claim 7 is shown in drawing 8 . Property verification equipment is replaced with the single pass expression converter 212 and the procedure transform-processing section 213 shown in drawing 5 in drawing 8 , it has the transform-processing section 216 equivalent to the transform-processing means 123 stated by claim 7, it changes into the procedure train which mentioned above the count tree logical expression received through the

input reception section 211, and it has become with the configuration with which processing of the set-operation section 214 is presented.
[0056] In this transform-processing section 216, the formula deformation processing section 228 has the composition of changing count tree logical expression into a procedure train, according to the transformation rule held at reception and the transformation-rule storing section 229 in the count tree logical expression with which the property for verification is expressed through the input reception section 211. This transformation-rule storing section 229 stores the transformation rule T expressed like a formula (19) to a formula (23), using the subexpression f of the propositional-logic types p and q and count tree logical expression as transformation rule from count tree logical expression to a procedure train.
[0057]
T (p) =p ... (19) T (p**EX f) =FindTrans (p) **T (f) ... (20) T (p**EF f) =FindTrail(p, True) **T (f) ... (21) T (p**EG q) =FindLoop (FindTrail (p**q, q) (**q) ... (22) T ()) [p**E (q U f) =FindTrail(p, q) **T(f)] ... (23), therefore the formula deformation processing section 241 detect the subexpression which suits the transformation rule mentioned above from the received count tree logical expression, and should just apply the transformation rule applicable to each subexpression one by one.
[0058] In this case, drawing 6 (a) As shown, it is once denied by the input reception section 211. Moreover, propositional-logic type pi about an initial state It responds to the input of the added count tree logical expression. The formula deformation processing section 228 operates and it is drawing 6 (b). By applying the transformation rule shown in the subexpression shown by the underline which set and attached the sign b1 and the sign b2 by the formula (21) and the formula (22), respectively, it is drawing 6 (c) directly from count tree logical expression. The shown procedure train can be acquired.
[0059] Thus, it becomes possible to realize the property verification equipment which applied the method of claim 3. In this case, by changing count tree logical expression into a procedure train directly, it becomes possible to reduce the procedures which constitute a property verification activity, and it becomes possible to shorten further the processing time which the whole verification activity takes.
[0060] Next, how to verify also including the property which is not possible for a single pass expression is explained. The property which is not possible for a single pass expression is expressed with the count tree logical expression containing the subexpression which is not convertible for a procedure train by the transformation rule R mentioned above and transformation rule S, and in order to verify the property expressed with this whole count tree logical expression, it needs to apply reverse image count about such a subexpression at least.
[0061] Here, a reverse image count result is expressed with the propositional-logic type showing a condition set. Therefore, if the reverse image count about the subexpression which is not convertible is preceded and processed and this subexpression is replaced by the propositional-logic formula showing a reverse image count result Conversion in the procedure train which mentioned count tree logical expression above as a whole can be enabled, and the result of an operation equivalent to the case where it asks by the conventional method according to the original count tree logical expression is obtained by performing the set operation using image count according to this procedure train.
[0062] The operation gestalt of the property verification equipment of claim 7 which applied claim 8 to drawing 9 is shown. In the property verification equipment shown in drawing 9 , while the substitute processing section 217 replaces the subexpression which is not convertible for the procedure train included in the count tree logical expression received through the input reception section 211 by the suitable propositional-logic type and presenting processing of the transform-processing section 216, it has become with the configuration of presenting processing of the reverse image computation section 218 with the corresponding subexpression.
[0063] Detecting the subexpression which cannot apply the transformation rule mentioned above from the count tree logical expression which received the subexpression detecting element 231 in the substitute processing section 217 shown in drawing 9 , the sign allocation section 232 has the composition of replacing the corresponding subexpression with the sign which shows a meaning propositional-logic type in count tree logical expression, and presenting processing of the deformation

processing section 216 with it. Here, the subexpression detecting element 231 should just detect subexpressions other than the tense logical expression shown by the left part of a formula to a formula mentioned above.
[0064] Moreover, the propositional-logic type in which is asked for the condition set shown by the subexpression which the reverse image computation section 218 received using reverse image count as usual, and the synthetic processing section 219 shows this condition set receives, and it replaces with the corresponding sign which is contained in a procedure train, it substitutes in drawing 9 , and it has become with the configuration which offers to processing of the set-operation section 214 in the acquired procedure train.
[0065] Here, the propositional-logic type itself does not change in transform processing to the single pass expression from count tree logical expression, and transform processing from a single pass expression to a procedure train. Therefore, by changing into a procedure train the count tree logical expression which transposed the subexpression which cannot be changed as mentioned above to the sign which shows a propositional-logic type, and replacing the sign contained in this procedure train by the propositional-logic formula showing a reverse image count result In the logical expression input means 111, this subexpression can be replaced by the propositional-logic type obtained by performing reverse image computation about the corresponding subexpression, and a result equivalent to the procedure train which presents transform processing with this count tree logical expression, and is acquired can be obtained.
[0066] That is, the subexpression detecting element 231 of the substitute processing section 217 and the reverse image computation section 218 can realize the function of the separation means 125 stated by claim 8, and the reverse image count means 126 in this case, respectively, and verification of the count tree logical expression which realizes the function of the synthetic means 127 stated by claim 7, and contains the subexpression which is not convertible can carry out by the sign allocation section 244 and the synthetic processing section 219 as it is possible.
[0067] Hereafter, actuation of the property verification equipment shown in drawing 9 is explained about a concrete example. since a head is the tense operator E which shows "existence" when count tree logical expression "EG** (p**AGq)" is inputted into the input reception section 211 -- drawing 10 (a) Propositional-logic type pi showing an initial state in count tree logical expression as it is so that it may be shown It is added and processing of the substitute processing section 217 is presented.
[0068] According to this, the subexpression detecting element 231 of the substitute processing section 217 detects a subexpression "AGq" as a subexpression which cannot apply the transformation rule T mentioned above, and the sign allocation section 232 assigns sign q' which shows a propositional-logic type to the subexpression "AGq" received from the subexpression detecting element 231, and should just replace it by it (refer to drawing 10 (b)). After applying the transformation rule which showed the whole count tree logical expression in the formula (22) by this at the count tree logical expression with which it became convertible into a procedure train, and the deformation processing section 216 was obtained according to the input of this count tree logical expression, the procedure train corresponding to (drawing 10 (b) reference) and count tree logical expression is acquired by arranging a formula further.
[0069] Moreover, when the reverse image computation section 218, on the other hand, performs the same reverse image computation as usual according to the input of a subexpression "AGq" mentioned above, the propositional-logic type showing the condition set expressed with a subexpression "AGq" is called for, and is inputted into the synthetic processing section 219 with the procedure train mentioned above. It responds to this and the synthetic processing section 219 is drawing 10 (b) about this propositional-logic type. By substituting for sign q' contained in the shown procedure train, and presenting processing by the set operation section 214, the condition set which fills the property expressed with the original count tree logical expression can be asked, and processing of the judgment processing section 215 is presented with this condition set.
[0070] Thus, it becomes possible to realize the property verification equipment which can verify all the properties expressed with count tree logical expression by applying the method of claim 4 and applying

reverse image computation only about the subexpression which is not convertible for a procedure train. In this case, since the count tree logical expression showing a property is a part very much, that reverse image computation is applied can shorten substantially the processing time spent on reverse image computation compared with the conventional method which applies reverse image computation to the whole count tree logical expression.

[0071] Therefore, as it mentioned above, in a realistic computer system, it becomes possible using the symbolic model inspection technique to verify all properties, and the property verification equipment which can verify the model of a logical unit logically from various viewpoints can be realized. Since this becomes possible [not leaking and discovering the inconvenience of the specification etc. in the earliest phase of layout of a logical unit,] and the acquired information can be fed back to layout, layout of large-scale LSI, a computing system, etc. can be supported powerfully, and the increase in efficiency can be promoted.

[0072]

[Effect of the Invention] Since it becomes possible to verify about the property by which the single pass expression was carried out only using image count according to the invention method of claim 1, and the invention equipment of claim 5 using this method as explained above, in the symbolic model inspection technique, it is compatible in the control of the amount of memory and the compaction of computation time which are obtained by function-izing state-transition relation.

[0073] It becomes possible to realize property verification by the symbolic model inspection technique using the computer system which had a realistic throughput by this, and in the earliest phase of layout of a logical unit, verification which does not have leakage about most properties can be performed, this verification result can be fed back to layout, and layout of a logical unit can be supported.

[0074] Furthermore, if invention of claim 2 and claim 6 is applied, it will become possible to perform efficient symbolic model verification about the thing in which a single pass expression is possible among the properties expressed with the count tree logical expression generally used as an expression of a property. Moreover, if invention of claim 3 and claim 7 is applied, it is possible to shorten further the time amount which reduces procedures required for a verification activity and processing takes by changing the inputted count tree logical expression into a procedure train directly.

[0075] Moreover, when applying invention of claim 4 and claim 8 and the subexpression **** count tree logical expression which needs reverse image computation is inputted as a property for verification on the other hand, reverse image computation is selectively applied to the corresponding subexpression. After changing into a propositional-logic type, by verifying the whole count tree logical expression, it becomes possible to apply to verification of all the properties that can be expressed with count tree logical expression, and the versatility of the property verification method and equipment can be improved substantially.

[Translation done.]

* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

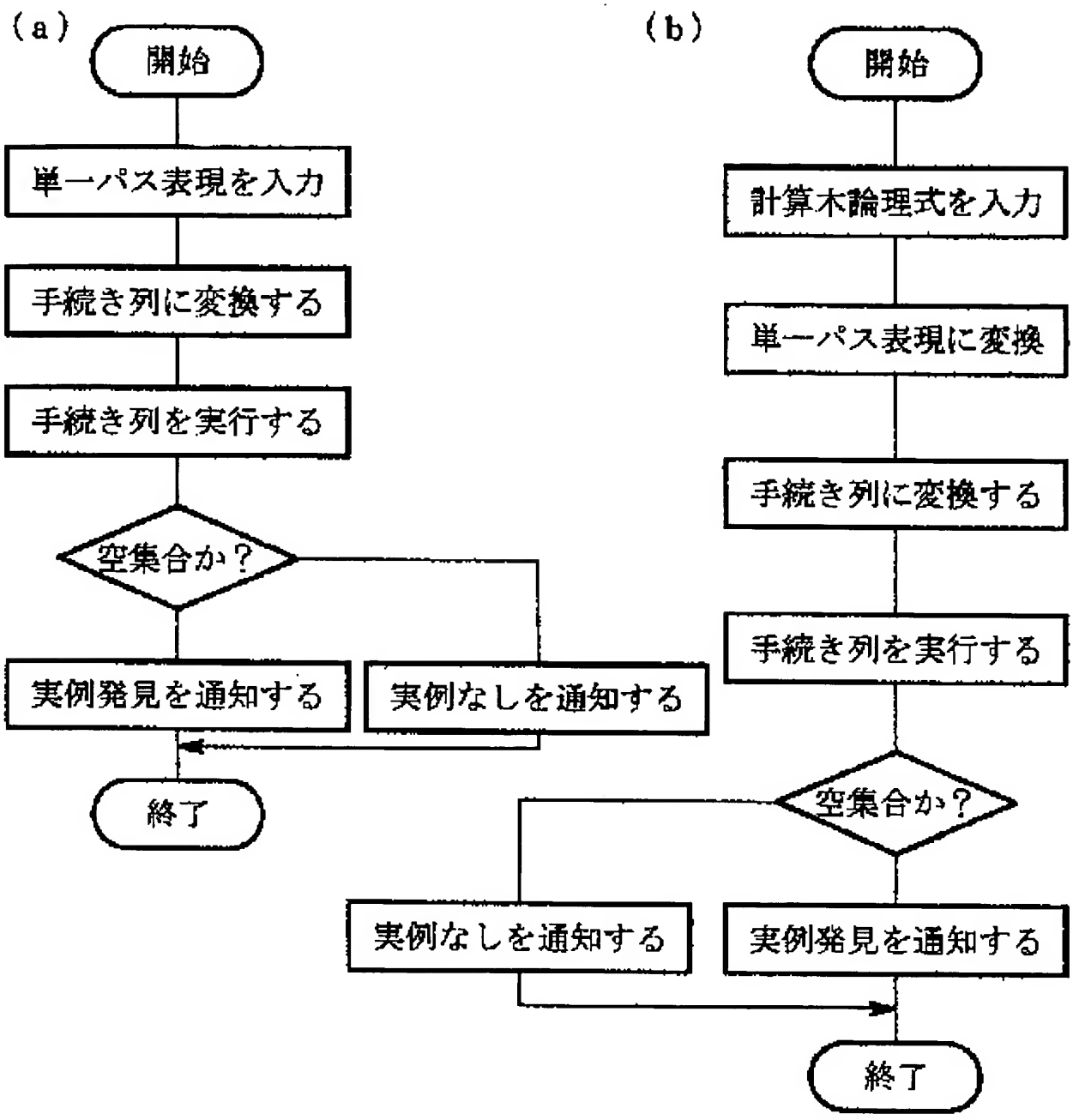
DESCRIPTION OF DRAWINGS

[Brief Description of the Drawings]
[Drawing 1] It is drawing showing the principle of the property verification method of claim 1 and claim 2.
[Drawing 2] It is drawing showing the principle of the property verification method of claim 3 and claim 4.
[Drawing 3] It is the principle block diagram of the property verification equipment of claim 5 and claim 6.
[Drawing 4] It is the principle block diagram of the property verification equipment of claim 7 and claim 8.
[Drawing 5] It is drawing showing the operation gestalt of the property verification equipment of claim 6.
[Drawing 6] It is drawing explaining transform processing to a procedure train.
[Drawing 7] It is drawing explaining a basic procedure.
[Drawing 8] It is drawing showing the operation gestalt of the property verification equipment of claim 7.
[Drawing 9] It is drawing showing the operation gestalt of the property verification equipment of claim 6 which applied invention of claim 8.
[Drawing 10] It is drawing explaining transform processing to a procedure train.
[Drawing 11] It is drawing showing the example of a finite state machine.
[Drawing 12] It is drawing explaining the processing time of the conventional verification procedure.
[Description of Notations]
111 Single Pass Expression Input Means
112 Procedure Train Conversion Means
113 Data-Processing Means
114 Judgment Means
121 Logical Expression Input Means
122 Expression Conversion Means
123 Transform-Processing Means
124 Reception Means
125 Separation Means
126 Reverse Image Count Means
127 Synthetic Means
211 Input Reception Means
212 Single Pass Expression Converter
213 Procedure Train Converter
214 Set Operation Section
215 Judgment Processing Section
216 Transform-Processing Section

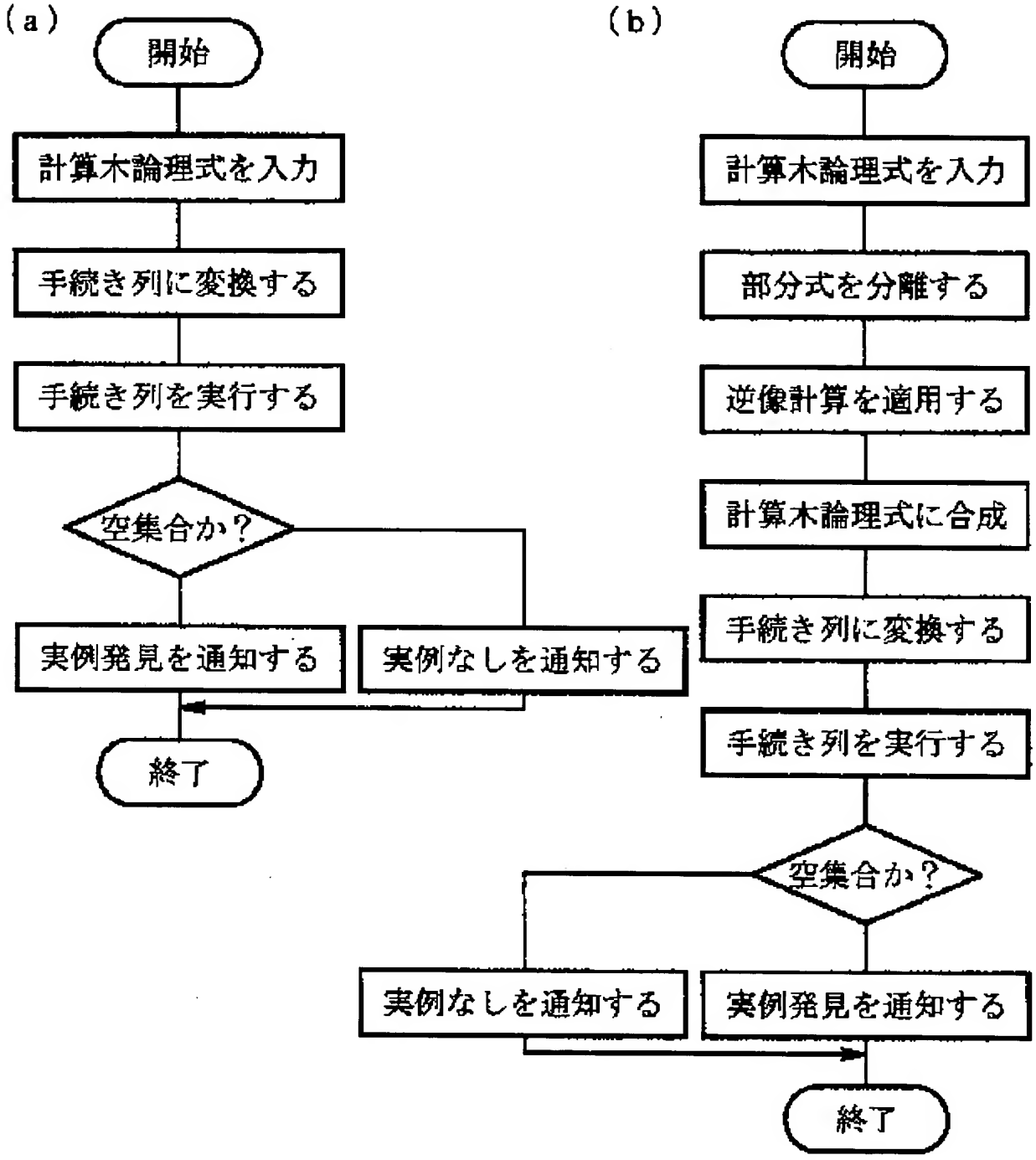
217 Substitute Processing Section
218 Reverse Image Computation Section
219 Synthetic Processing Section
221 Operator Distinction Section
222 Negative Processing Section
223 Initial-State Adjunct
224, 226, 228 Formula deformation processing section
225, 227, 229 Transformation-rule storing section
231 Subexpression Detecting Element
232 Sign Allocation Section

[Translation done.]

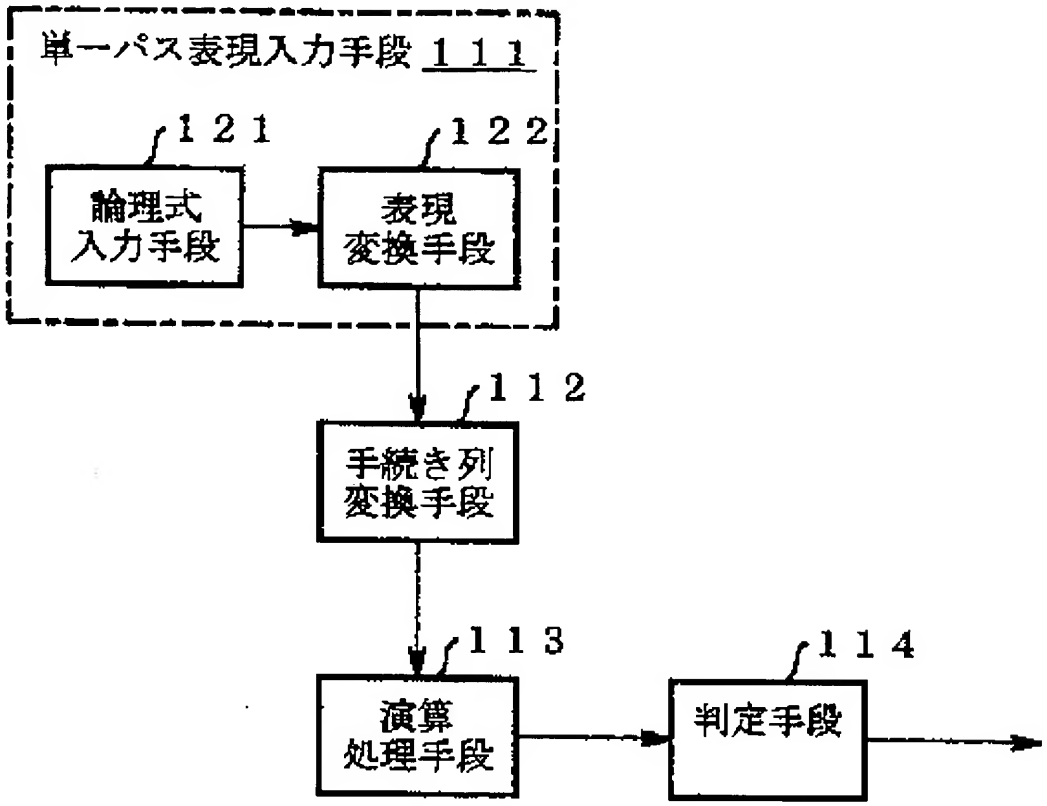
請求項１および請求項２のプロパティ検証方法の原理を示す図



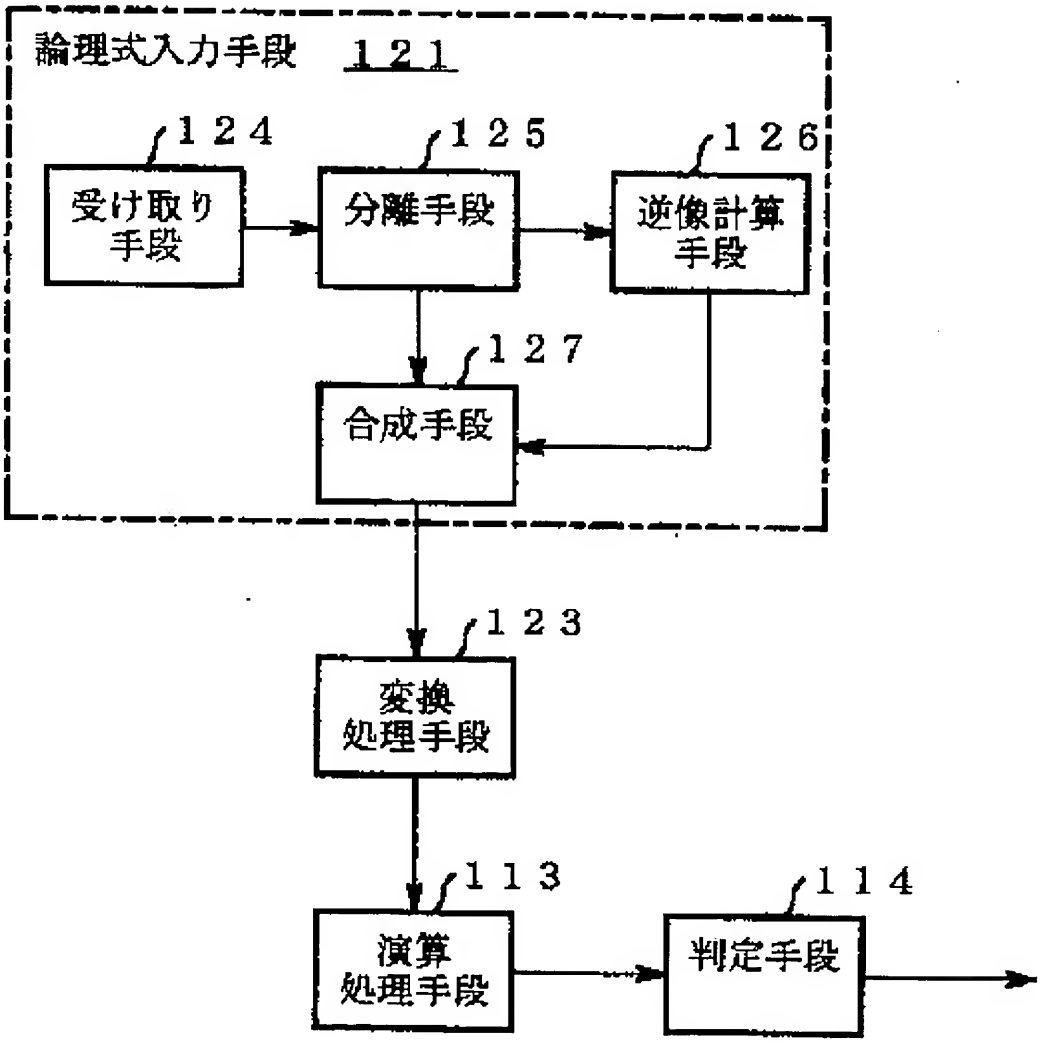
請求項３および請求項４のプロパティ検証方法の原理を示す図



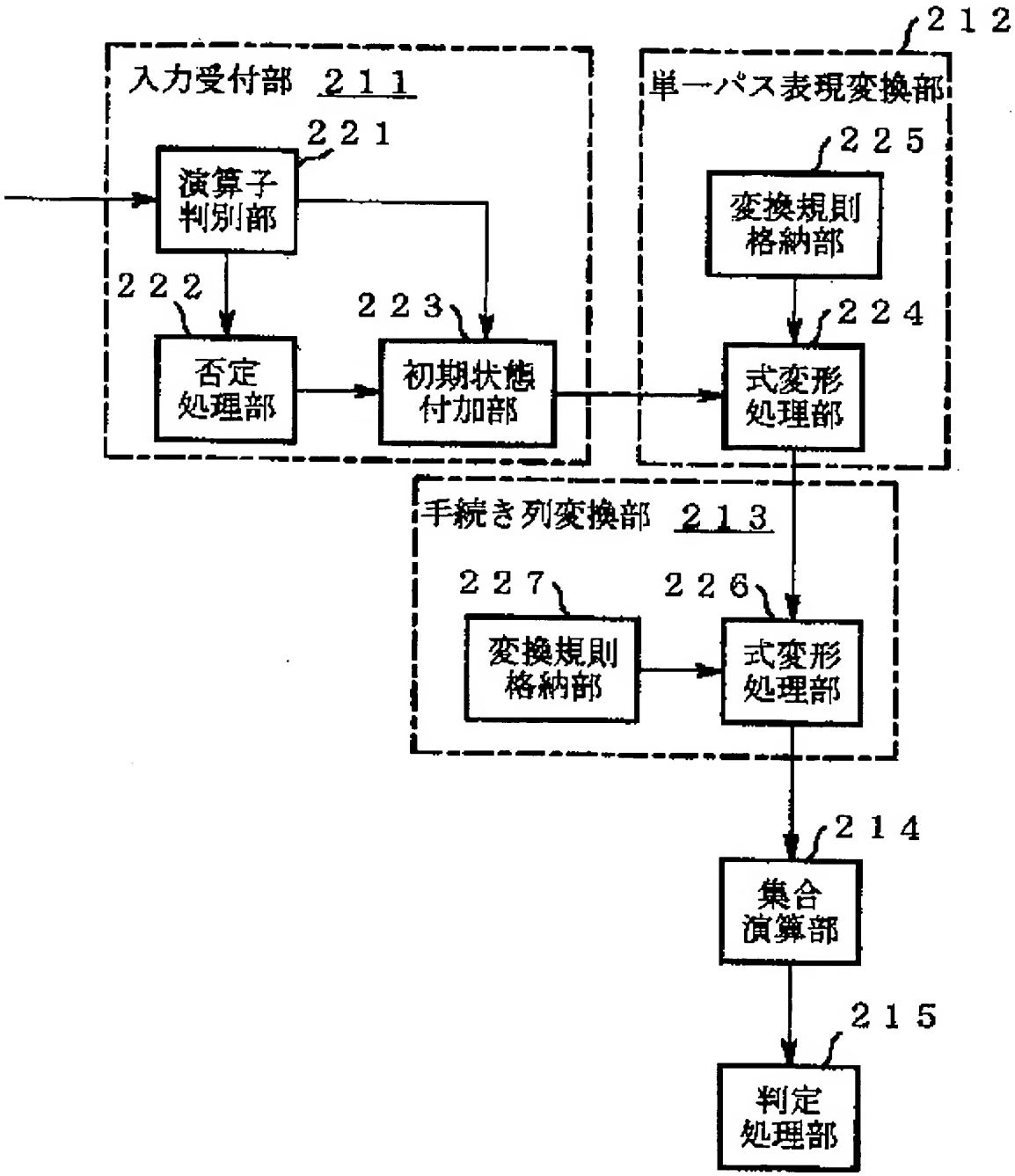
請求項５および請求項６のプロパティ検証装置の原理ブロック図



請求項７および請求項８のプロパティ検証装置の原理ブロック図



請求項６のプロパティ検証装置の実施形態を示す図



手続き列への変換処理を説明する図

(a) $\neg AG\ (p \rightarrow AF\ q) = \neg AG\ (\neg p \vee AF\ q)$

$= EF \neg\ (\neg p \vee AF\ q)$

$= EF\ (p \wedge \neg AF\ q)$

$= EF\ (p \wedge EG \neg q)$

(b) $\underbrace{pi \wedge EF}_{b1}\underbrace{(p \wedge EG \neg q)}_{b2}$

$\rightarrow \{pi \wedge R\ (p \wedge EG \neg q),\ pi\ (True)^{*}\ R\ (p \wedge EG \neg q)\}$

$\rightarrow \{(pi \wedge p \wedge \neg q), pi\ (True)^{*}\ (p \wedge \neg q)\}\ (\neg q)^{\omega}$

(c) $\{(pi \wedge p \wedge \neg q), pi\ (True)^{*}\ (p \wedge \neg q)\}\ (\neg q)^{\omega}$

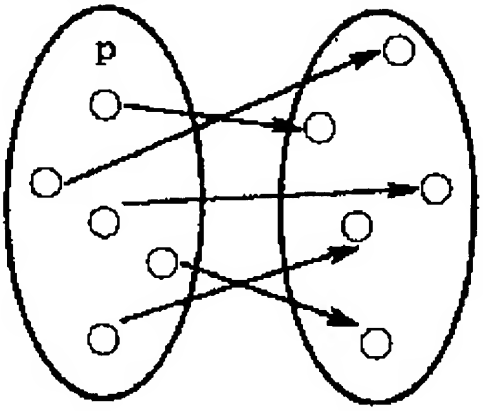
$\rightarrow$ FindLoop(FindTrail(S({pi∧p∧¬q), pi(True)* (p∧q)}), ¬q) ∧¬q)

$\rightarrow$ FindLoop(FindTrail(FindTrail(pi, (True)) ∧(p∧¬q), ¬q) ∧¬q)

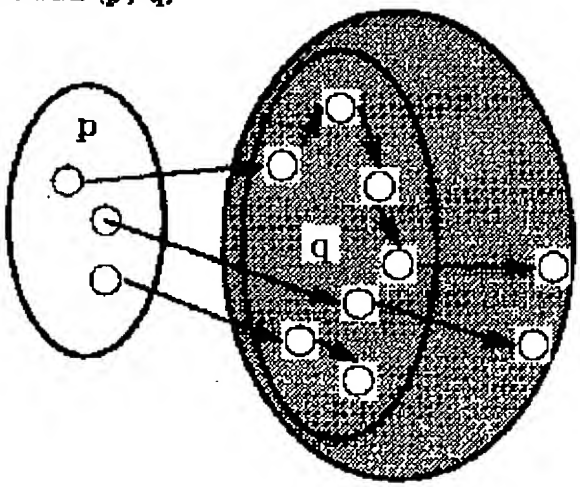
$\rightarrow$ FindLoop(FindTrail(FindTrail(pi, (True) ∧p, ¬q) ∧¬q)

基本手続きを説明する図

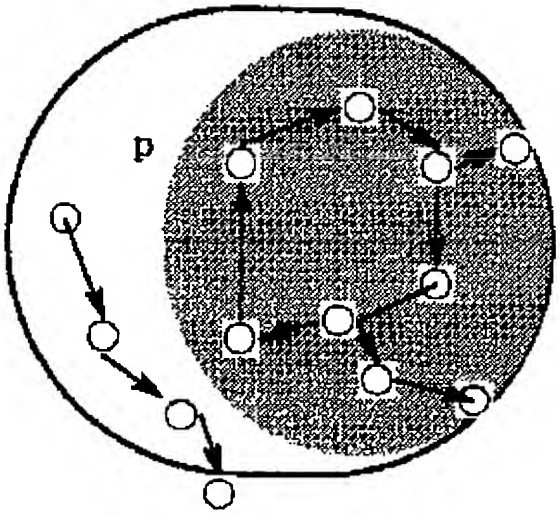
(a) FindTrans(p)



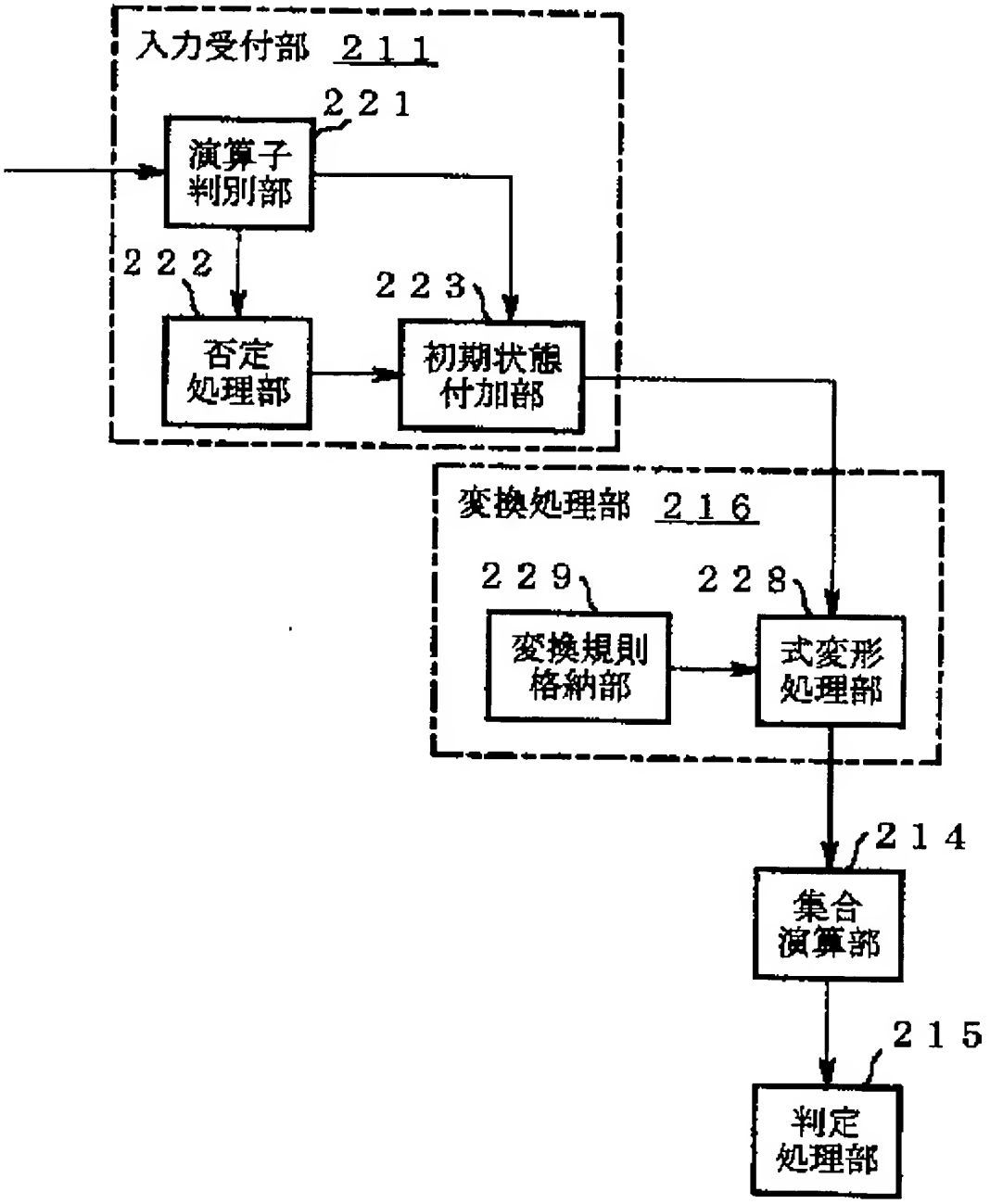
(b) FindTrail(p, q)



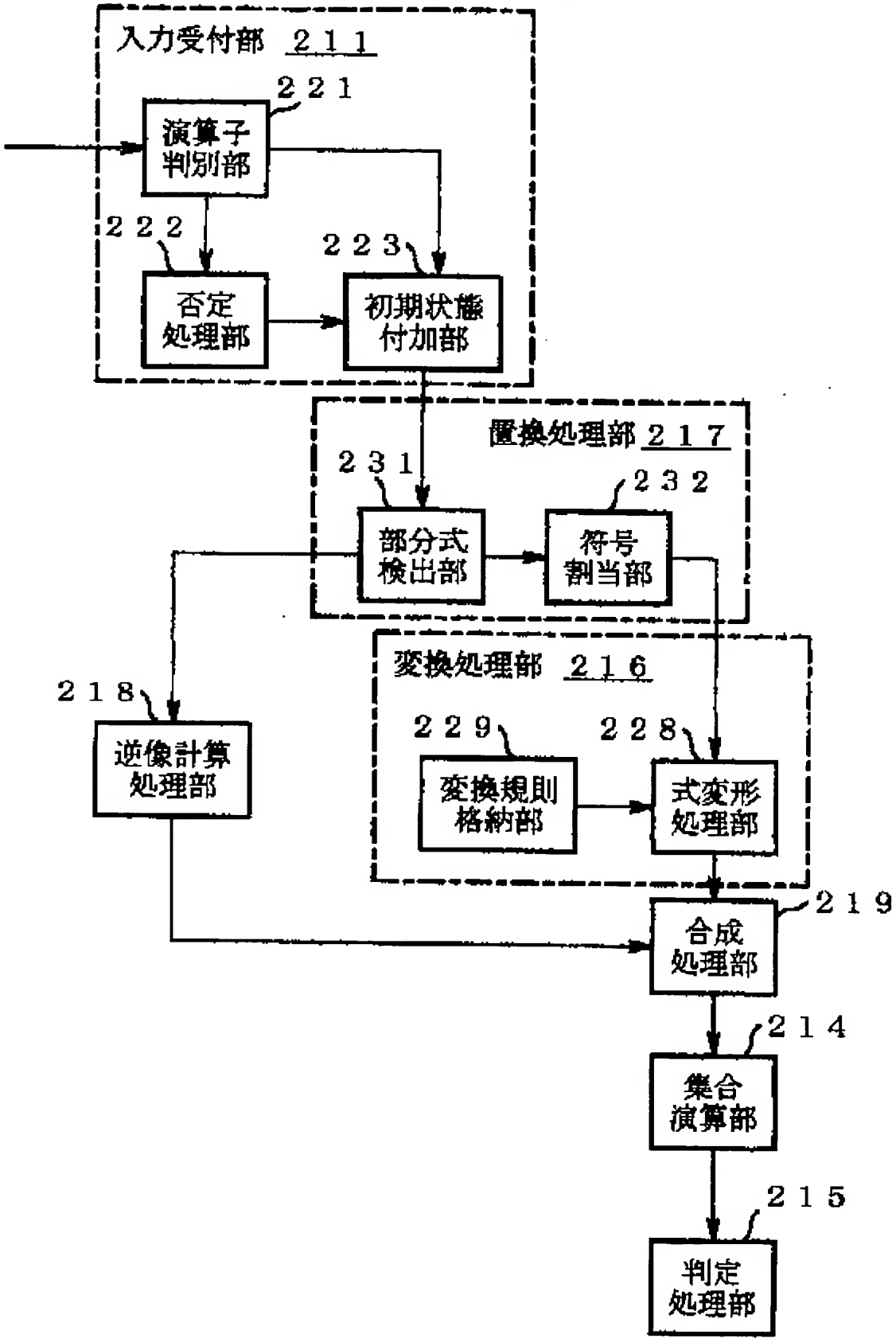
(c) FindLoop(p)



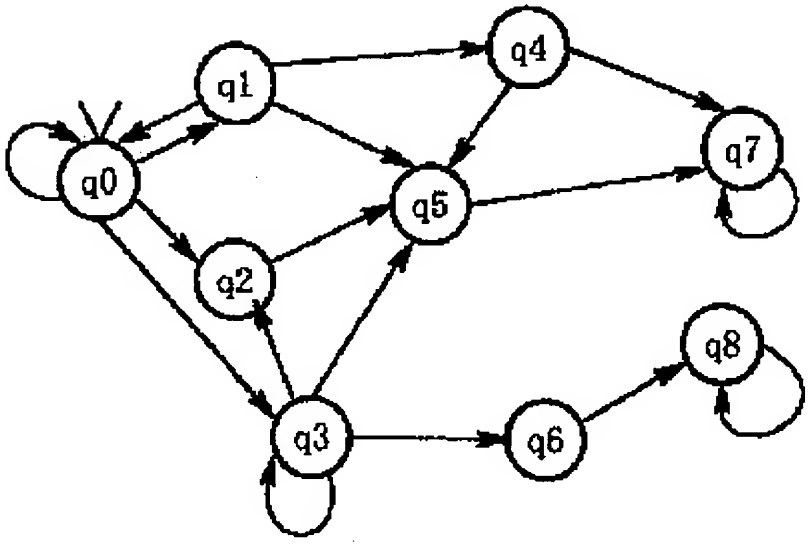
請求項７のプロパティ検証装置の実施形態を示す図



請求項８のプロパティ検証装置の実施形態を示す図



有限状態機械の例を示す図



手続き列への変換処理を説明する図

(a)

$$pi \wedge EG\ (p \wedge AG\ q) \rightarrow pi \wedge EG\ (p \wedge q')$$

(b)

$$pi \wedge EG\ (p \wedge q')$$

$$\rightarrow \mathrm{FindLoop}(\mathrm{FindTrail}((pi \wedge p \wedge q'), p \wedge q') \wedge (p \wedge q')$$

検証手続きの処理時間を説明する図

モデル	関数化なし		関数化あり	
	像計算	逆像計算	像計算	逆像計算
atm-sw	8.9	21.8	10.9	202.5
dh-1	0.0	0.8	0.3	1.3
dh-2	13.8	75.4	46.8	>10000
vpp	space	space	3.0	4135.1
pipe-s	10.7	14.8	0.6	387.9
pipe-d	space	space	200.4	>20000

(19)日本国特許庁（ＪＰ）　(12) 公開特許公報（Ａ）　(11)特許出願公開番号

特開平10－63537

(43)公開日　平成10年(1998)３月６日

(51)Int.Cl.[6]		識別記号	庁内整理番号	Ｆ Ｉ			技術表示箇所
Ｇ０６Ｆ	11/28	３４０		Ｇ０６Ｆ	11/28	３４０Ａ	
	17/50				15/60	６６４Ｇ	

審査請求　未請求　請求項の数８　ＯＬ　（全 13 頁）

(21)出願番号　特願平8－220005

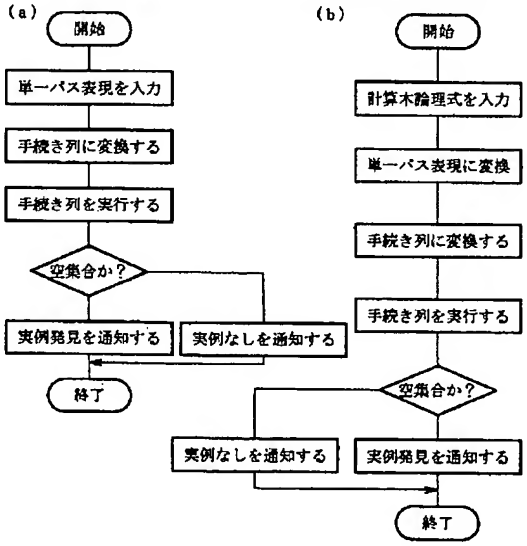
(22)出願日　平成８年(1996)８月21日

(71)出願人　000005223
富士通株式会社
神奈川県川崎市中原区上小田中４丁目１番１号
(72)発明者　中田　恒夫
神奈川県川崎市中原区上小田中４丁目１番１号　富士通株式会社内
(72)発明者　岩下　洋哲
神奈川県川崎市中原区上小田中４丁目１番１号　富士通株式会社内
(74)代理人　弁理士　古谷　史旺　（外１名）

(54)【発明の名称】　プロパティ検証方法および装置

(57)【要約】
【課題】　記号モデル検査法において、メモリ規模の縮小と処理時間の短縮とを両立可能なプロパティ検証方法および装置を提供する。
【解決手段】　同期式順序機械の動作を表す有限状態機械が、機能仕様を表すプロパティを満たしているか否かを検証するプロパティ検証方法において、検証対象のプロパティとして、枝分かれのない一本の状態集合列あるいは枝分かれのない一本の状態集合列に無限ループが連結した構成の状態集合列を示す単一パス表現が入力され、単一パス表現を有限状態機械における像計算処理のみで実行可能な所定の手続きを順次に組合せた手続き列に変換し、手続き列で示された各手続きを実行して、単一パス表現に対応する状態遷移のパスを経由する状態集合を求め、得られた状態集合が空集合であるか否かに基づいて、実例発見あるいは実例なしを通知する。

請求項１および請求項２のプロパティ検証方法の原理を示す図



1

【特許請求の範囲】

【請求項１】　同期式順序機械の動作を表す有限状態機械が、機能仕様を表すプロパティを満たしているか否かを検証するプロパティ検証方法において、

検証対象のプロパティとして、枝分かれのない一本の状態集合列あるいは枝分かれのない一本の状態集合列に無限ループが連結した構成の状態集合列を示す単一パス表現が入力され、

前記単一パス表現を前記有限状態機械における像計算処理のみで実行可能な所定の手続きを順次に組合せた手続き列に変換し、

前記手続き列で示された各手続きを実行して、前記単一パス表現に対応する状態遷移のパスを経由する状態集合を求め、

得られた状態集合が空集合であるか否かに基づいて、実例発見あるいは実例なしを通知することを特徴とするプロパティ検証方法。

【請求項２】　請求項１に記載のプロパティ検証方法において、

検証対象のプロパティを表す情報として、計算木論理式が入力され、

所定の変換規則に従って、前記計算木論理式を単一パス表現に変換し、手続き列への変換処理に供することを特徴とするプロパティ検証方法。

【請求項３】　同期式順序機械の動作を表す有限状態機械が、機能仕様を表すプロパティを満たしているか否かを検証するプロパティ検証方法において、

検証対象のプロパティを表す情報として、計算木論理式が入力され、

所定の変換規則に従って、前記計算木論理式を前記有限状態機械における像計算処理のみで実行可能な所定の手続きを順次に組合せた手続き列に変換し、

前記手続き列で示された各手続きを実行して、前記単一パス表現に対応する状態遷移のパスを経由する状態集合を求め、

得られた状態集合が空集合であるか否かに基づいて、実例発見あるいは実例なしを通知することを特徴とするプロパティ検証方法。

【請求項４】　請求項２または請求項３に記載のプロパティ検証方法において、

所定の条件に基づいて、入力された計算木論理式を逆像計算の適用が必要な部分式とその他の計算木論理式とに分離し、

前記部分式について逆像計算処理を実行して、状態集合列を示す命題論理式を求め、

前記命題論理式と前記その他の計算木論理式とを合成して、後段の処理に供することを特徴とするプロパティ検証方法。

【請求項５】　同期式順序機械の動作を表す有限状態機械が、機能仕様を表すプロパティを満たしているか否か

2

を検証するプロパティ検証装置において、

検証対象のプロパティとして、枝分かれのない一本の状態集合列あるいは枝分かれのない一本の状態集合列に無限ループが連結した構成の状態集合列を示す単一パス表現を入力する単一パス表現入力手段と、

前記単一パス表現を所定の手続きを順次に組合せた手続き列に変換する手続き列変換手段と、

前記手続き列で示された各手続きに従って、前記有限状態機械における像計算処理を実行して、前記単一パス表現に対応する状態遷移のパスを経由する状態集合を求める演算処理手段と、

得られた状態集合が空集合であるか否かを判定する判定手段とを備えたことを特徴とするプロパティ検証装置。

【請求項６】　請求項５に記載のプロパティ検証装置において、

単一パス表現入力手段は、

検証対象のプロパティを表す計算木論理式を入力する論理式入力手段と、

所定の変換規則に従って、前記計算木論理式を単一パス表現に変換し、手続き列変換手段の処理に供する表現変換手段とを備えた構成であることを特徴とするプロパティ検証装置。

【請求項７】　同期式順序機械の動作を表す有限状態機械が、機能仕様を表すプロパティを満たしているか否かを検証するプロパティ検証装置において、

検証対象のプロパティを表す計算木論理式を入力する論理式入力手段と、

所定の変換規則に従って、前記計算木論理式を所定の手続きを順次に組合せた手続き列に変換する変換処理手段と、

前記手続き列で示された各手続きに従って、前記有限状態機械における像計算処理を実行して、前記単一パス表現に対応する状態遷移のパスを経由する状態集合を求める演算処理手段と、

得られた状態集合が空集合であるか否かを判定する判定手段とを備えたことを特徴とするプロパティ検証装置。

【請求項８】　請求項６または請求項７に記載のプロパティ検証装置において、

論理式入力手段は、

プロパティを表す計算木論理式の入力を受け取る受け取り手段と、

前記計算木論理式を逆像計算処理の適用が必要とされる部分式とその他の計算木論理式とに分離する分離手段と、

前記部分式に対して逆像計算処理を適用し、状態集合を示す命題論理式を求める逆像計算手段と、

前記命題論理式と前記その他の計算木論理式とを合成し、プロパティを表す計算木論理式として出力する合成手段とを備えた構成であることを特徴とするプロパティ検証装置。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は、有限状態機械で与えられた論理装置のモデルが、論理装置の機能仕様を満たすか否かを検証するプロバティ検証方法および装置に＊関するものである。有限状態機械は、出力を持つ有限オートマトンとして定義され、状態の集合Ｑ、入力アルファベットΣ、出力アルファベットΦ、遷移関係式δ、出力関係式Λおよび初期状態集合Ｉの６項組で式（１）のように表される。

$$M=(Q,\ \Sigma,\ \Phi,\ \delta,\ \Lambda,\ I) \quad \cdots (1)$$

上述した遷移関係式δは、非決定的な遷移を表すために、現状体、次状態および入力が与えられたときに、現状態においてその入力が与えられたときに次状態に遷移する場合があれば値として１を取り、なければ値として０を取ることにより、次の状態を決める関数である。ここで、全ての同期式順序回路は、原則的に有限状態機械でモデル化できるため、論理装置を設計する際には、その仕様として有限状態機械を用いる方法が広く用いられている。例えば、論理合成による回路設計においては、設計記述言語で表された装置の仕様を有限状態機械に変換し、各状態をフリップフロップやレジスタで実現し、また、遷移関係式や出力関係式を組み合わせ回路で実現する方法が一般的である。このようにして設計された論理装置のモデルの検証作業は、抽象的な仕様に関する検証作業から現実の回路の動作に関する検証作業まで、様々な段階から構成されている。これらの段階うち、機能仕様の検証は、論理装置の設計の最も早い段階における誤り検出が期待できるため、迅速かつ確実な検証方法が必要とされている。

【０００２】

【従来の技術】論理装置の機能仕様を検証する手法としては、大きく分けて、論理シミュレーション手法と形式的検証手法とが挙げられる。論理シミュレーション手法は、論理装置のモデルに対して適切な入力を加えて論理装置の動作を模擬し、得られた出力が元の有限状態機械から得られる出力と一致するか否か調べることにより、論理装置のモデルの正当性を検証する手法である。

【０００３】したがって、論理シミュレーション手法では、様々な状況を考慮して入力データを作成する作業が必要不可欠であり、また、この入力データの作成作業において考慮されなかった状況に関しては検証することができない。これに対して、形式的検証手法は、論理装置のモデルが元の有限状態機械が満たしているプロバティを実現しているか否かを数学的に検証するものである。

【０００４】なお、形式的検証に関しては、例えば、「論理関数処理に基づく形式的検証手法」平石、浜口、情報処理、Vol.35(8):pp710-718,1994（以下、文献１と称する）などの文献を参照されたい。形式的検証手法としては、プロバティを時相論理の一種である計算木論理で表現する記号モデル検査手法と、無限の入力列に対応すべく受理条件を拡張したω－オートマトンによってプロバティを表現する言語包含検査手法（R.Hojati,R.K.B※rayton,and R.P.Kurshan. "BDD-based debugging of designs using languagecontainment and fair CTL." In Proceedings of the Conference on ComputerAided Verification,1993）が提案されている。

【０００５】形式的検証手法のうち記号モデル検査手法は、Kripke構造の論理装置モデルを論理関数を用いて表現し、計算木論理で表された仕様を満足する空でない状態集合が存在するか否かを検査することにより、論理装置のモデルが仕様を満足するか否かを検証する手法である。ここで、Kripke構造Ｋとは、状態の有限集合Ｓと状態間の遷移関係Ｒと初期状態点の集合Ｓ。と各状態で真となる原始命題の集合Ｌとを用いて、式（２）のように表される非決定的有限オートマトンの一種である。

【０００６】

$$K=(S,\ R,\ S_0,\ L) \quad \cdots (2)$$

また、計算木論理は、時相論理の一種であり、通常の論理演算子に加えて、全称を表す演算子Ａ、存在を表す演算子Ｅとともに、「いつか」を表す時相演算子Ｆ、「いつも」を表す時相演算子Ｇ、「次に」を表す時相演算子Ｘおよび「まで」を表す時相演算子Ｕを用いて表される。

【０００７】例えば、時相論理ＡＧａは、初期状態から到達可能な状態集合の全てにおいて論理式ａが成立する旨を示している。この場合は、論理装置のモデルにおいて、初期状態から遷移可能な全ての道筋を辿っていき、その道筋の全てが論理式ａが成立している状態に辿り着くか否かを調べればよい。

【０００８】すなわち、記号モデル検査手法における検証作業は、Kripke構造の状態遷移を辿り、各状態において、仕様を表す計算木論理式が成立するか否かを確認する作業であり、この作業は、計算木論理の式を用いたモデル上の最小不動点あるいは最大不動点を求める集合演算に帰着される。このような集合演算は、ある状態集合$\{q\}$から１回の遷移で到達可能な状態集合を求める像計算$Image(\{q\})$と、ある状態集合$\{q\}$に１回の状態遷移で到達可能な状態集合を求める逆像計算$Image^{-1}(\{q\})$とを組み合わせることによって実現される。

【０００９】例えば、図１１に示すように、９つの状態q_0～q_8間の状態遷移で表される有限状態機械の例において、像計算および逆像計算を行った結果の例を式（３）から式（６）に示す。

$$Image(\{q_0\})=\{q_0,q_1,q_2,q_3\} \quad \cdots (3)$$

$$Image(\{q_0,q_8\})=\{q_2,q_3,q_5,q_6,q_8\} \quad \cdots (4)$$

$$Image^{-1}(\{q_0\})=\{q_0,q_1\} \quad \cdots (5)$$

$$Image^{-1}(\{q_3\})=\{q_1,q_2,q_3,q_4\} \quad \cdots (6)$$

この図１１に示した有限状態機械において、例えば、状態ｑ，を示す論理式ｐを用いた時相論理ＡＦ ｐを検証する際には、初期状態ｑ。から順次に像計算を繰り返し、初期状態から遷移可能な全ての道筋が状態ｑ，に到達するか否かを調べればよい。一方、時相論理ＥＦ ｐを検証する場合には、状態ｑ，から逆像計算を繰り返して、初期状態ｑ。に到達する道筋があるか否かを調べればよい。

【００１０】ところで、実際の記号モデル検査手法においては、集合演算を論理関数処理に置き換える手法が一般的であり（文献１参照）、また、論理関数を二分決定グラフで表現することにより、論理関数処理の効率化が図られている（R.E.Bryant."Graph based algorithm for boolean function manipulation." IEEE Trans.Comput.,C－35(8):pp677-691,1986.）。

【００１１】また、状態遷移関係の非決定的な部分に新たな入力を加えて制御することにより、次状態を現状態と入力（元の入力と新たな入力）との関数として表し、状態遷移関係を表す二分決定グラフの規模の縮小を図る手法も提案されている。

【００１２】

【発明が解決しようとする課題】上述したように、二分決定グラフによって非決定的な遷移関係を表した場合は、図１２の「関数化なし」の欄に示すように、検証手続きとして用いられる像計算と逆像計算とをほぼ同じオーダーの時間で処理可能である。

【００１３】しかしながら、この場合には、二分決定グラフの規模を示すノード数は、入力変数に依存しており、最悪の場合には、ノード数が入力変数の増大に応じて指数関数的に増大する。このため、実験用のモデルの中でも状態数の大きい論理装置のモデルに適用した場合には、二分決定グラフを格納するために膨大なメモリ容量を必要とするために、そのままでは現実的な計算機システムで扱うことはほとんど不可能だった。

【００１４】図１２においては、300MB 以上のメモリ容量を必要としたために検証不能となったモデル（例えば、vpp,pipe-1）については、該当する欄に符号spaceを付して示した。一方、状態遷移関係を関数で表し、二分決定グラフの規模の縮小を図った場合には、図１２の「関数化あり」の欄に示すように、二分決定グラフの規模を現実的な計算機システムで処理可能な大きさに抑えることができる反面、逆像計算に要する処理時間が著しく増大する。その一方、記号モデル検査手法においては、像計算よりもむしろ逆像計算の方が、検証作業において多用されているため、逆像計算に要する時間の増大により、検証作業全体に要する時間が大幅に増大し、現実的な計算機システムの処理能力では扱えなくなってしまう。

【００１５】このため、従来の手法をそのまま用いたのでは、必要とされるメモリの規模あるいは必要とされる処理時間が爆発し、現実のプロセッサなどに相当する規模の論理装置を検証することがほとんどできなかった。

【００１６】ところで、上述した従来の手法においては、いずれも、計算木論理によって表現可能なプロパティの全てに対応して、検証作業を可能とすることを前提としている。しかしながら、論理装置において検証したいプロパティは、計算木論理によって表現可能なプロパティのごく一部であることが経験的に分かっている。

【００１７】例えば、時相論理式「ＥＦ ｐ」は、「初期状態から論理式ｐが成り立つ状態集合Ｑp に到達可能である」旨を示しており、論理式ｐにより、論理装置が陥ってはならない禁止状態を示して、論理装置のモデルが禁止状態に陥らない旨のプロパティを検証する場合などに頻繁に用いられる。また、時相論理式「ＡＧ(p→ＡＦ q）」は、「論理式ｐが成り立つ状態集合のうち、初期状態から到達可能な部分集合から、どのようなパスを通ったとしても必ず論理式ｑが成り立つ状態集合を通る」旨を示しており、例えば、リクエストが発生したら必ず応答が発生する旨のプロパティを検証する場合などに頻繁に用いられる。

【００１８】上述した時相論理式「ＥＦ ｐ」の実例は、初期状態から状態集合Ｑp に至る状態集合列となり、また一方、時相論理式の「ＡＧ(p→ＡＦ q）」反例は、初期状態から論理式ｐが成り立つ状態集合を通った後、常に、論理式¬ｑが成り立つループに至る状態集合列となる。これらの状態集合列は、いずれも枝分かれのない一本の状態集合列あるいはこのような一本の状態集合列の末尾に無限ループとなる状態集合列が連結した構成となっている。以下、このような状態集合列の構成を単一パスと称し、また、単一パス構成の状態集合列によって、実例あるいは反例を表現することが可能なプロパティを単一パス表現のプロパティと称する。

【００１９】ところで、発明者らの推定によれば、上述したような単一パス表現のプロパティは、論理装置のモデルにおいて検証作業の対象となるプロパティの９０パーセントから９５パーセントを占めていると考えられる。したがって、単一パス表現のプロパティに関する検証を効率化することができれば、論理装置モデルの検証作業全体を大幅に効率化することが可能である。

【００２０】本発明は、記号モデル検査法において、メモリ規模の縮小と処理時間の短縮とを両立可能なプロパティ検証方法および装置を提供することを目的とする。

【００２１】

【課題を解決するための手段】図１(a),(b) は、請求項１および請求項２のプロパティ検証方法の原理を示す図である。

【００２２】請求項１の発明は、同期式順序機械の動作を表す有限状態機械が、機能仕様を表すプロパティを満たしているか否かを検証するプロパティ検証方法において、検証対象のプロパティとして、枝分かれのない一本の状態集合列あるいは枝分かれのない一本の状態集合列に無限ループが連結した構成の状態集合列を示す単一パス表現が入力され、単一パス表現を有限状態機械における像計算処理のみで実行可能な所定の手続きを順次に組合せた手続き列に変換し、手続き列で示された各手続きを実行して、単一パス表現に対応する状態遷移のパスを経由する状態集合を求め、得られた状態集合が空集合であるか否かに基づいて；実例発見あるいは実例なしを通知することを特徴とする。

【００２３】請求項１の発明は、入力された単一パス表現を変換して得られた手続き列に従って、像計算処理を行うことによりプロパティを満たす状態集合を求め、この状態集合が空集合であるか否かに基づいて、プロパティを検証することができる。すなわち、検証対象のプロパティが単一パス表現可能であれば、像計算処理のみを利用して、その検証作業を実現することができるから、記号モデル検査法において、状態遷移関係の関数化による二分決定グラフの規模縮小を適用したことによるメモリ規模の縮小と処理時間の短縮とを両立することが可能となる。

【００２４】請求項２の発明は、請求項１に記載のプロパティ検証方法において、検証対象のプロパティを表す情報として、計算木論理式が入力され、所定の変換規則に従って、計算木論理式を単一パス表現に変換し、手続き列への変換処理に供することを特徴とする。請求項２の発明は、計算木論理式を単一パス表現に変換して、手続き列への変換処理に供することができるから、計算木論理式で表されたプロパティの検証に用いることが可能となる。

【００２５】図２(a),(b) は、請求項３および請求項４のプロパティ検証方法の原理を示す図である。請求項３の発明は、同期式順序機械の動作を表す有限状態機械が、機能仕様を表すプロパティを満たしているか否かを検証するプロパティ検証方法において、検証対象のプロパティを表す情報として、計算木論理式が入力され、所定の変換規則に従って、計算木論理式を有限状態機械における像計算処理のみで実行可能な所定の手続きを順次に組合せた手続き列に変換し、手続き列で示された各手続きを実行して、単一パス表現に対応する状態遷移のパスを経由する状態集合を求め、得られた状態集合が空集合であるか否かに基づいて、実例発見あるいは実例なしを通知することを特徴とする。

【００２６】請求項３の発明は、計算木論理式を直接に手続き列に変換する過程を考えることにより、計算機論理式で表されたプロパティの検証に必要な手順を削減することができ、単一パス表現可能なプロパティ検証作業を更に効率化することができる。請求項４の発明は、請求項２または請求項３に記載のプロパティ検証方法において、所定の条件に基づいて、入力された計算木論理式を逆像計算の適用が必要な部分式とその他の計算木論理式とに分離し、部分式について逆像計算処理を実行して、状態集合列を示す命題論理式を求め、命題論理式とその他の計算木論理式とを合成して、後段の処理に供することを特徴とする。

【００２７】請求項４の発明は、計算木論理式の一部の部分式についてのみ逆像計算処理を適用し、その結果として得られる命題論理式を元の計算機論理式に合成して、プロパティの検証処理に供することにより、検証対象のプロパティについての制限を解除することができる。これにより、計算木論理式で表現可能な全てのプロパティを検証対象とすることが可能となる。

【００２８】図３は、請求項５および請求項６のプロパティ検証装置の原理を示す図である。請求項５の発明は、同期式順序機械の動作を表す有限状態機械が、機能仕様を表すプロパティを満たしているか否かを検証するプロパティ検証装置において、検証対象のプロパティとして、枝分かれのない一本の状態集合列あるいは枝分かれのない一本の状態集合列に無限ループが連結した構成の状態集合列を示す単一パス表現を入力する単一パス表現入力手段１１１と、単一パス表現を所定の手続きを順次に組合せた手続き列に変換する手続き列変換手段１１２と、手続き列で示された各手続きに従って、有限状態機械における像計算処理を実行して、単一パス表現に対応する状態遷移のパスを経由する状態集合を求める演算処理手段１１３と、得られた状態集合が空集合であるか否かを判定する判定手段１１４とを備えたことを特徴とする。

【００２９】請求項５の発明は、手続き列変換手段１１２により、単一パス表現入力手段１１１を介して受け取った単一パス表現を手続き列に変換し、この手続き列に従って演算処理手段１１３が動作し、得られた状態集合を判定手段１１４の処理に供することにより、請求項１で述べたプロパティ検証方法を適用して、単一パス表現で示されるプロパティを検証することができる。

【００３０】これにより、単一パス表現可能なプロパティについては、像計算処理のみを利用して、その検証作業を実現することができるから、記号モデル検査法において、状態遷移関係の関数化による二分決定グラフの規模縮小を適用したことによるメモリ規模の縮小と処理時間の短縮とを両立することが可能となる。

【００３１】請求項６の発明は、請求項５に記載のプロパティ検証装置において、単一パス表現入力手段１１１は、検証対象のプロパティを表す計算木論理式を入力する論理式入力手段１２１と、所定の変換規則に従って、計算木論理式を単一パス表現に変換し、手続き列変換手段１１２の処理に供する表現変換手段１２２とを備えた

構成であることを特徴とする。

【００３２】請求項６の発明は、論理式入力手段１２１による計算機論理式の入力に応じて、表現変換手段１２２が動作することにより、計算木論理式で表されたプロバティを検証することができる。図４は、請求項７および請求項８のプロバティ検証装置の原理を示す図である。請求項７の発明は、同期式順序機械の動作を表す有限状態機械が、機能仕様を表すプロバティを満たしているか否かを検証するプロバティ検証装置において、検証対象のプロバティを表す計算木論理式を入力する論理式入力手段１２１と、所定の変換規則に従って、計算木論理式を所定の手続きを順次に組合せた手続き列に変換する変換処理手段１２３と、手続き列で示された各手続きに従って、有限状態機械における像計算処理を実行して、単一パス表現に対応する状態遷移のパスを経由する状態集合を求める演算処理手段１１３と、得られた状態集合が空集合であるか否かを判定する判定手段１１４とを備えたことを特徴とする。

【００３３】請求項７の発明は、変換処理手段１２３が、論理式入力手段１２１を介して受け取った計算木論理式を直接に手続き列に変換し、単一パス表現を経由する必要性を排除することにより、計算木論理式で表されたプロバティの検証作業を更に迅速に行うことが可能となる。請求項８の発明は、請求項６または請求項７に記載のプロバティ検証装置において、論理式入力手段１２１は、プロバティを表す計算木論理式の入力を受け取る受け取り手段１２４と、計算木論理式を逆像計算処理の適用が必要とされる部分式とその他の計算木論理式とに分離する分離手段１２５と、部分式に対して逆像計算処理を適用し、状態集合を示す命題論理式を求める逆像計算手段１２６と、命題論理式とその他の計算木論理式とを合成し、プロバティを表す計算木論理式として出力する合成手段１２７とを備えた構成であることを特徴とする。

【００３４】請求項８の発明は、分離手段１２５が、受け取り手段１２４を介して受け取った計算木論理式の一部を分離して逆像計算手段１２６の処理に供し、合成手段１２７により、逆像計算で得られた命題論理式を元の計算木論理式に合成することにより、逆像計算が必要な部分式を含む計算木論理式を単一パス表現可能な計算木論理式に変換することができる。

【００３５】これにより、検証対象となるプロバティに関する制限を解除し、あらゆるプロバティを表す計算木論理式の入力に応じて検証作業を行うことができるから、汎用性に優れたプロバティ検証装置を実現すること＊ができる。

【００３６】

【発明の実施の形態】以下、図面に基づいて、本発明の実施形態について詳細に説明する。

【００３７】図５は、請求項６のプロバティ検証装置の実施形態を示す図である。図５において、プロバティ検証装置は、論理式入力手段１２１に相当する入力受付部２１１を介して受け付けた計算木論理式を単一パス表現変換部２１２によって単一パス表現に変換し、更に、手続き列変換部２１３によって後述する手続き列に変換した後に集合演算部２１４の処理に供し、得られた状態集合が空であるか否かを判定処理部２１５によって判定することにより、入力されたプロバティを検証する構成となっている。

【００３８】図５に示した入力受付部２１１において、演算子判別部２２１は、受け取った計算木論理式の先頭の演算子に応じて、この計算木論理式を直接あるいは否定処理部２２２を介して初期状態付加部２２３に送出する構成となっている。ここで、演算子判別部２２１は、先頭の演算子が全称を示す演算子「Ａ」である場合に、計算木論理式全体を否定処理部２２２に送出し、一方、存在を示す演算子「Ｅ」である場合には、そのまま初期状態付加部２２３に送出すればよい。

【００３９】また、図５において、否定処理部２２２は、演算子判別部２２１から受け取った計算木論理式全体を否定し、更に、論理演算規則に従って必要な式の変形を行って計算木論理式を整理して、得られた計算木論理式を初期状態付加部２２３に送出する構成となっている。この初期状態付加部２２３は、受け取った計算木論理式の先頭に初期状態に対応する命題論理式piを付加することにより、検証作業の開始時点を初期状態とし、この計算木論理式を単一パス表現変換部２１２に送出する構成となっている。

【００４０】また、図５に示した単一パス表現変換部２１２において、式変形処理部２２４は、上述した入力受付部２１１から計算木論理式を受け取り、変換規則格納部２２５に保持された変換規則に従ってこの計算木論理式を変形し、単一パス表現に変換する構成となっている。ここで、変換規則格納部２２５には、計算木論理式を単一パス表現に変換する手続きＲとして、状態集合を示す命題論理式ｐ，ｑおよび上述した計算木論理式の部分式ｆを用いて式（７）から式（10）に示す変換規則を格納しておけばよい。

【００４１】

$$R(p \wedge EX\ f) = pR(f) \quad \cdots (7)$$

$$R(p \wedge EF\ f) = \{R(p \wedge f),\ p\,True^{*}\ R(f)\} \quad \cdots (8)$$

$$R(p \wedge EG\ q) = (p \wedge f)\ q\omega \quad \cdots (9)$$

$$R(p \wedge E(q\ U\ f)) = \{R(p \wedge f),\ p\,q^{*}\ R(f)\} \quad \cdots (10)$$

式（７）から式（10）においては、命題論理式ｐの有限回および無限回の繰り返しを指数符号「＊」および指数

符号「ω」で示し、また、場合分けを中括弧でくくって示している。

【００４２】また、式変形処理部２２４は、受け取った計算木論理式から上述した変換規則に適合する箇所を検出し、該当する変換規則をそれぞれ適用することにより、単一パス表現への変換処理を行えばよい。

【００４３】ここで、単一パス表現とは、枝分かれのない一本の状態集合列あるいは、枝分かれのない一本の状態集合列に無限ループとなる状態集合列が連結した状態集合列であり、任意の状態集合を表す命題論理式の集合Ｐを用いて、単一パス表現ｓの集合Ｓを次のように定義する。ｐ∈Ｐである状態集合ｐは、単一パス表現である。

【００４４】ｐ∈Ｐである状態集合ｐとｓ∈Ｓである状態集合列ｓとの連接ｐｓは、単一パス表現である。ｐ∈Ｐである状態集合ｐの無限回の繰り返しｐω は、単一パス表現である。ｐ∈Ｐである状態集合ｐの有限回の繰り返しｐ* とｓ∈Ｓである状態集合列ｓとの連接ｐ* ｓは、単一パス表現である。

【００４５】したがって、式変形処理部２２４は、上述した変換規則に従って式を変形していき、以上で定義した単一パス表現で表される状態集合列を求めればよい。例えば、入力受付部２１１を介して検証すべきプロパティを示す計算木論理式「ＡＧ（ｐ→ＡＦｑ）」が入力された場合は、先頭の演算子が全称を示す演算子「Ａ」であるので、演算子判別部２２１からの指示に応じて否定処理部２２２が動作し、図６(a) に示すように、入力された計算木論理式を一旦否定した後に整理することによ＊り、演算子「Ａ」を含まない計算木論理式が得られる。更に、初期状態付加部２２３により、この計算木論理式の先頭に、初期状態に対応する命題論理式ｐｉ が付加され、得られた計算木論理式が式変形処理部２２４の処理に供される。

【００４６】この計算木論理式の入力に応じて、式変形処理部２２４が動作し、図６(b) において、それぞれ斜線を付して示す部分式に、上述した式（８）および式（９）を順次に適用して整理することにより、図６(b) に示す単一パス表現が得られる。このように、単一パス表現変換部２１２の各部が動作することにより、請求項６で述べた表現変換手段１２２の機能を実現し、入力受付部２１１を介して受け取った計算木論理式を単一パス表現に変換し、全体として、請求項５で述べた単一パス表現入力手段１１１の機能を実現し、手続き列変換手段１１２に相当する手続き列変換部２１３の処理に供することができる。

【００４７】図５に示した手続き列変換部２１３において、式変形処理部２２６は、変換規則格納部２２７に保持された変換規則に従って、単一パス表現を以下に述べる検証手続き列に変換する構成となっている。ここで、基本的な検証手続き列は、像計算のみによって構成されており、命題論理式ｐ、ｑと像計算を示す演算子Img および最大不動点と最小不動点をそれぞれ表す演算子gfp 、lfp を用いて、式(11)から式(13)のように表される。

【００４８】

$$\mathrm{FindTrans}(p) = \mathrm{Img}(p) \quad \cdots (11)$$

$$\mathrm{FindTrail}(p,q) = \mathrm{lfp}\ Z.[p \vee \mathrm{Img}(q \wedge Z)] \quad \cdots (12)$$

$$\mathrm{FindLoop}(p) = \mathrm{gfp}\ Z.[p \wedge \mathrm{Img}(Z)] \quad \cdots (13)$$

式(11)に示した検証手続きFindTrans(p)は、まさに、像計算そのものであり、図７(a) に示すように、命題論理式ｐで表された状態集合から一回の状態遷移で到達可能な状態集合を返す手続きである。

【００４９】また、式(12)に示した検証手続きFindTrail(p,q)は、図７(b) に示すように、命題論理式ｐで表された状態集合から、命題論理式ｑが成り立つ状態集合を経由して到達可能な状態集合（図７(b) において斜線を付して示す）を返す手続きである。また、式(13)に示し※た検証手続きFindLoop(p) は、図７(c) に示すように、命題論理式ｐで表された状態集合に含まれており、かつ、命題論理式ｐが成り立つループを構成する部分集合とそのループから到達可能な部分集合の和集合（図７(c) において斜線を付して示す）を返す手続きである。

【００５０】また、変換規則格納部２２７は、単一パス表現ｓおよび命題論理式ｐを用いて式(14)～式(18)のように表される変換規則Ｓを格納している。

$$S(p) = p \quad \cdots (14)$$

$$S(sp) = \mathrm{FindTrans}(S(s)) \wedge p \quad \cdots (15)$$

$$S(sp^*) = \mathrm{FindTrail}(\mathrm{FindTrans}(S(s)), p) \quad \cdots (16)$$

$$S(sp\omega) = \mathrm{FindLoop}(\mathrm{FindTrail}(S(s), p) \wedge p) \quad \cdots (17)$$

$$S(\{s \wedge p, sp^*\}) = \mathrm{FindTrail}(S(s), p) \quad \cdots (18)$$

したがって、式変形処理部２２６は、受け取った単一パス表現からこれらの変換規則に適合する部分を検出し、該当する変換規則を順次に適用することにより、単一パス表現を検証手続き列に変換すればよい。

【００５１】例えば、図６(b) に示した単一パス表現の入力に応じて、式変形処理部２２６が式(17)および式(18)を順次に適用し、更に式を整理することにより、図６(c)に示すようにして、手続き列に変換することができる。このようにして得られた手続き列の入力に応じて、集合演算部２１４は、上述した式(11)から式(13)に示し

た像計算および不動点計算を実行し、単一パス表現で表される状態集合を求め、判定処理部２１５に入力し、状態集合が空集合であるか否かを判定すればよい。

【００５２】このようにして、請求項１の方法を適用したプロバティ検証装置を実現することができる。上述したように、これらの手続き列は像計算のみを用いて実行可能であるから、状態遷移関係を関数で表して、二分木決定グラフの規模を縮小した場合においても、現実的な計算機システムの処理能力を用いて集合演算処理を実行し、論理装置のモデルが、単一パス表現で表すことが可能なプロバティを実現しているか否かを検証することができる。

【００５３】これにより、単一パス表現可能なプロバティに関しては、記号モデル検査手法を適用する際に問題となるメモリ量の抑制と処理時間の短縮とを両立することができるので、記号モデル検査手法を現実的な規模を持つ論理装置のプロバティ検証に適用することが可能となる。

【００５４】ここで、上述したように、現実の論理装置において検証が必要となるプロバティのほとんどは単一パス表現可能であり、これらのプロバティの検証作業に本発明の技法を適用して検証作業の効率化を図ることに＊より、検証作業全体を大幅に効率化することができる。更に、計算木論理式を直接に上述した手続き列に変換することも可能である。

【００５５】図８に、請求項７のプロバティ検証装置の実施形態を示す。図８において、プロバティ検証装置は、図５に示した単一パス表現変換部２１２および手続き変換処理部２１３に代えて、請求項７で述べた変換処理手段１２３に相当する変換処理部２１６を備え、入力受付部２１１を介して受け取った計算木論理式を上述した手続き列に変換し、集合演算部２１４の処理に供する構成となっている。

【００５６】この変換処理部２１６において、式変形処理部２２８は、入力受付部２１１を介して検証対象のプロバティを表す計算木論理式を受け取り、変換規則格納部２２９に保持された変換規則に従って、計算木論理式を手続き列に変換する構成となっている。この変換規則格納部２２９は、計算木論理式から手続き列への変換規則として、命題論理式ｐ、ｑおよび計算木論理式の部分式ｆを用いて、式(19)から式(23)のように表される変換規則Ｔを格納している。

【００５７】

$$T(p) = p \quad \cdots(19)$$

$$T(p \wedge EX\ f) = \mathrm{FindTrans}(p) \wedge T(f) \quad \cdots(20)$$

$$T(p \wedge EF\ f) = \mathrm{FindTrail}(p,\mathrm{True}) \wedge T(f) \quad \cdots(21)$$

$$T(p \wedge EG\ q) = \mathrm{FindLoop}(\mathrm{FindTrail}((p \wedge q, q) \wedge q) \quad \cdots(22)$$

$$T(p \wedge E(q\ U\ f) = \mathrm{FindTrail}(p,q) \wedge T(f) \quad \cdots(23)$$

したがって、式変形処理部２４１は、受け取った計算木論理式から上述した変換規則に適合する部分式を検出し、それぞれの部分式に該当する変換規則を順次に適用していけばよい。

【００５８】この場合は、図６(a)に示したように、入力受付部２１１によって一旦否定され、また初期状態に関する命題論理式ｐｉが付加された計算木論理式の入力に応じて、式変形処理部２２８が動作し、図６(b)において符号b1および符号b2を付した下線で示した部分式にそれぞれ式(21)および式(22)で示した変換規則を適用することにより、計算木論理式から直接に図６(c)に示した手続き列を得ることができる。

【００５９】このようにして、請求項３の方法を適用したプロバティ検証装置を実現することが可能となる。この場合は、計算木論理式を直接に手続き列に変換することにより、プロバティ検証作業を構成する手順を削減することが可能となり、検証作業全体に要する処理時間を更に短縮することが可能となる。

【００６０】次に、単一パス表現可能でないプロバティも含めて検証する方法について説明する。単一パス表現可能でないプロバティは、上述した変換規則Ｒおよび変換規則Ｓによって手続き列に変換することはできない部分式を含んだ計算木論理式で表されており、この計算木論理式全体で表されるプロバティを検証するためには、少なくとも、このような部分式については逆像計算を適用する必要がある。

【００６１】ここで、逆像計算結果は状態集合を示す命題論理式で表される。したがって、変換できない部分式に関する逆像計算を先行して処理し、この部分式を逆像計算結果を示す命題論理式で置換すれば、計算木論理式を全体として上述した手続き列に変換可能とすることができ、この手続き列に従って像計算を用いた集合演算を実行することにより、元の計算木論理式に従って従来方式で求めた場合と同等の演算結果が得られる。

【００６２】図９に、請求項８を適用した請求項７のプロバティ検証装置の実施形態を示す。図９に示したプロバティ検証装置において、置換処理部２１７は、入力受付部２１１を介して受け取った計算木論理式に含まれる手続き列に変換できない部分式を適切な命題論理式に置換して、変換処理部２１６の処理に供するとともに、該当する部分式を逆像計算処理部２１８の処理に供する構成となっている。

【００６３】図９に示した置換処理部２１７において、部分式検出部２３１は、受け取った計算木論理式から上述した変換規則が適用できない部分式を検出し、符号割当部２３２は、該当する部分式を計算木論理式において

一意な命題論理式を示す符号で置換して、変形処理部２１６の処理に供する構成となっている。ここで、部分式検出部２３１は、上述した式から式の左辺で示された時相論理式以外の部分式を検出すればよい。

【００６４】また、図９において、逆像計算処理部２１８は、受け取った部分式で示される状態集合を従来と同様に逆像計算を用いて求め、また、合成処理部２１９は、この状態集合を示す命題論理式を受け取って、手続き列に含まれる該当する符号に代えて代入し、得られた手続き列を集合演算部２１４の処理に供する構成となっている。

【００６５】ここで、計算木論理式から単一パス表現への変換処理および単一パス表現から手続き列への変換処理においては命題論理式自体は変化しない。したがって、上述したようにして、変換できない部分式を命題論理式を示す符号に置き換えた計算木論理式を手続き列に変換し、この手続き列に含まれる符号を逆像計算結果を示す命題論理式で置き換えることにより、論理式入力手段１１１において、該当する部分式に関する逆像計算処理を行って得られる命題論理式によってこの部分式を置き換え、この計算木論理式を変換処理に供して得られる手続き列と同等の結果を得ることができる。

【００６６】すなわち、この場合は、置換処理部２１７の部分式検出部２３１と逆像計算処理部２１８とにより、請求項８で述べた分離手段１２５と逆像計算手段１２６の機能をそれぞれ実現し、また、符号割当部２４４と合成処理部２１９とにより、請求項７で述べた合成手段１２７の機能を実現して、変換できない部分式を含む計算木論理式の検証を可能とすることができる。

【００６７】以下、具体的な例について、図９に示したプロバティ検証装置の動作を説明する。例えば、計算木論理式「ＥＧ∧（ｐ∧ＡＧｑ）」が入力受付部２１１に入力された場合は、先頭が「存在」を示す時相演算子Ｅであるので、図１０(a) に示すように、計算木論理式にそのまま初期状態を示す命題論理式ｐｉが付加され、置換処理部２１７の処理に供される。

【００６８】これに応じて、置換処理部２１７の部分式検出部２３１は、上述した変換規則Ｔが適用できない部分式として部分式「ＡＧｑ」を検出し、また、符号割当部２３２は、部分式検出部２３１から受け取った部分式「ＡＧｑ」に、命題論理式を示す符号q'を割り当てて置換すればよい（図１０(b) 参照）。これにより、計算木論理式全体を手続き列に変換可能となり、この計算木論理式の入力に応じて、変形処理部２１６が、得られた計算木論理式に式(22)に示した変換規則を適用した後、更に式を整理することにより（図１０(b) 参照）、計算木論理式に対応する手続き列が得られる。

【００６９】また一方、上述した部分式「ＡＧｑ」の入力に応じて、逆像計算処理部２１８が従来と同様の逆像計算処理を行うことにより、部分式「ＡＧｑ」で表される状態集合を示す命題論理式が求められ、上述した手続き列とともに、合成処理部２１９に入力される。これに応じて、合成処理部２１９が、この命題論理式を図１０(b) に示した手続き列に含まれる符号q'に代入し、集合演算部２１４による処理に供することにより、元の計算木論理式で表されたプロバティを満たす状態集合を求めることができ、この状態集合が判定処理部２１５の処理に供される。

【００７０】このようにして、請求項４の方法を適用し、手続き列に変換することのできない部分式についてのみ逆像計算処理を適用することにより、計算木論理式で表される全てのプロバティを検証することが可能なプロバティ検証装置を実現することが可能となる。この場合に、逆像計算処理が適用されるのは、プロバティを表す計算木論理式のごく一部であるから、計算木論理式全体に逆像計算処理を適用する従来方式に比べて、逆像計算処理に費やされる処理時間を大幅に短縮することが可能である。

【００７１】したがって、上述したようにして、現実的な計算機システムにおいて、記号モデル検査手法を利用して、あらゆるプロバティを検証することが可能となり、論理装置のモデルを様々な観点から、論理的に検証することが可能なプロバティ検証装置を実現することができる。これにより、論理装置の設計作業の最も早い段階で、その仕様の不都合などを漏れなく発見することが可能となり、得られた情報を設計作業にフィードバックすることができるから、大規模なＬＳＩや計算機システムなどの設計作業を強力に支援し、その効率化を推し進めることができる。

【００７２】

【発明の効果】以上に説明したように、請求項１の発明方法およびこの方法を用いた請求項５の発明装置によれば、単一パス表現されたプロバティについて、像計算のみを用いて検証を行うことが可能となるから、記号モデル検査手法において、状態遷移関係を関数化することで得られるメモリ量の抑制と計算時間の短縮を両立することができる。

【００７３】これにより、現実的な処理能力を持った計算機システムを用いて、記号モデル検査手法によるプロバティ検証を実現することが可能となり、論理装置の設計作業の最も早い段階において、大部分のプロバティに関して漏れのない検証を行うことができ、この検証結果を設計作業にフィードバックし、論理装置の設計作業を支援することができる。

【００７４】更に、請求項２および請求項６の発明を適用すれば、プロバティの表現として一般的に用いられている計算木論理式で表されたプロバティのうち、単一パス表現可能なものについて、効率的な記号モデル検証を行うことが可能となる。また、請求項３および請求項７の発明を適用すれば、入力された計算木論理式を直接に

17

手続き列に変換することにより、検証作業のために必要な手順を削減し、処理に要する時間を更に短縮することが可能である。

【００７５】また一方、請求項４および請求項８の発明を適用すれば、逆像計算処理が必要な部分式含む計算木論理式が検証対象のプロパティとして入力された場合に、該当する部分式に選択的に逆像計算処理を適用して、命題論理式に変換してから計算木論理式全体の検証を行うことにより、計算木論理式で表現可能なあらゆるプロパティの検証に適用することが可能となり、プロパティ検証方法および装置の汎用性を大幅に向上することができる。

【図面の簡単な説明】

【図１】請求項１および請求項２のプロパティ検証方法の原理を示す図である。

【図２】請求項３および請求項４のプロパティ検証方法の原理を示す図である。

【図３】請求項５および請求項６のプロパティ検証装置の原理ブロック図である。

【図４】請求項７および請求項８のプロパティ検証装置の原理ブロック図である。

【図５】請求項６のプロパティ検証装置の実施形態を示す図である。

【図６】手続き列への変換処理を説明する図である。

【図７】基本手続きを説明する図である。

【図８】請求項７のプロパティ検証装置の実施形態を示す図である。

【図９】請求項８の発明を適用した請求項６のプロパティ検証装置の実施形態を示す図である。

【図１０】手続き列への変換処理を説明する図である。

【図１１】有限状態機械の例を示す図である。

18

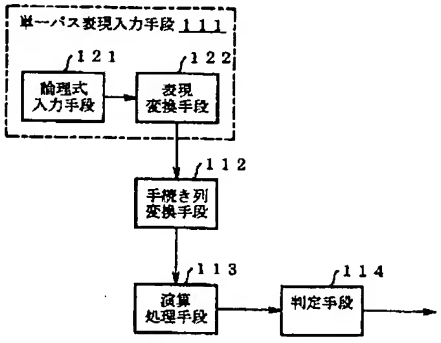
＊【図１２】従来の検証手続きの処理時間を説明する図である。

【符号の説明】

１１１　単一パス表現入力手段
１１２　手続き列変換手段
１１３　演算処理手段
１１４　判定手段
１２１　論理式入力手段
１２２　表現変換手段
１２３　変換処理手段
１２４　受け取り手段
１２５　分離手段
１２６　逆像計算手段
１２７　合成手段
２１１　入力受付手段
２１２　単一パス表現変換部
２１３　手続き列変換部
２１４　集合演算部
２１５　判定処理部
２１６　変換処理部
２１７　置換処理部
２１８　逆像計算処理部
２１９　合成処理部
２２１　演算子判別部
２２２　否定処理部
２２３　初期状態付加部
２２４、２２６、２２８　式変形処理部
２２５、２２７、２２９　変換規則格納部
２３１　部分式検出部
２３２　符号割当部

【図３】

請求項５および請求項６のプロパティ検証装置の原理ブロック図



【図１０】

手続き列への変換処理を説明する図

(a)

$$pi \wedge EG(p \wedge AGq) \rightarrow pi \wedge EG(p \wedge q')$$

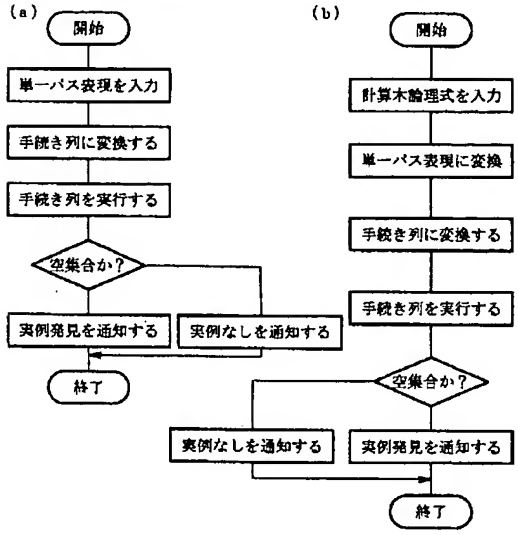
(b)

$$pi \wedge EG(p \wedge q')$$

$$\rightarrow FindLoop(FindTrail((pi \wedge p \wedge q'), p \wedge q') \wedge (p \wedge q')$$

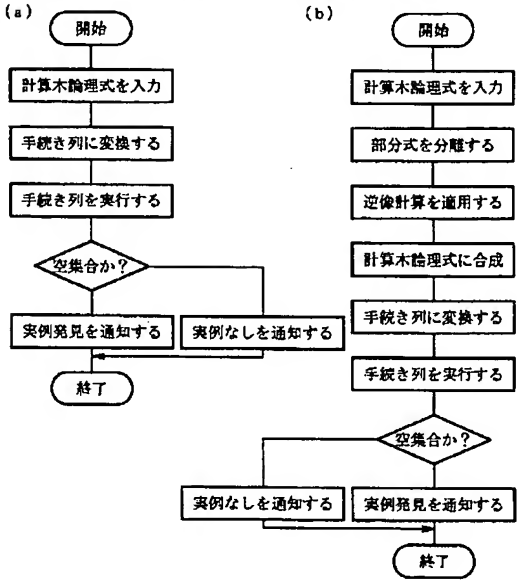
【図１】

請求項１および請求項２のプロパティ検証方法の原理を示す図



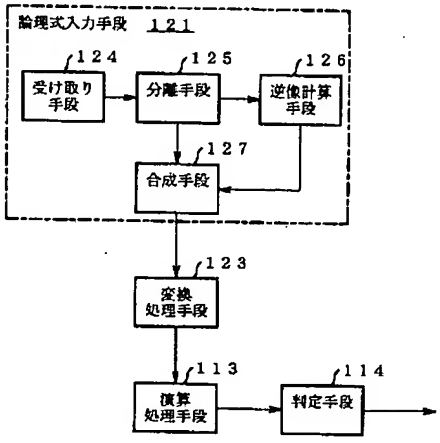
【図２】

請求項３および請求項４のプロパティ検証方法の原理を示す図



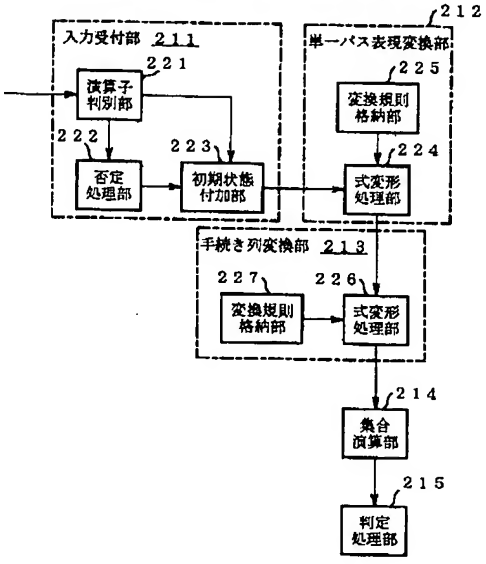
【図４】

請求項７および請求項８のプロパティ検証装置の原理ブロック図



【図５】

請求項６のプロパティ検証装置の実施形態を示す図



【図６】

手続き列への変換処理を説明する図

(a)　¬AG (p→AF q) =¬AG (¬p∨AF q)

=EF¬ (¬p∨AF q)

=EF (p∧¬AF q)

=EF (p∧EG¬q)

(b)　$\underbrace{pi \wedge EF(p \wedge EG \neg q)}_{b1 \quad b2}$

→{pi∧R (p∧EG¬q), pi(True)* R (p∧EG¬q)}

→{(pi∧p∧¬q), pi(True)* (p∧¬q)} (¬q)*

(c)　{(pi∧p∧¬q), pi(True) *(p∧¬q)} (¬q)*

→ FindLoop(FindTrail(S ({pi∧p∧¬q), pi(True)* (p∧q)}), ¬q)∧¬q)

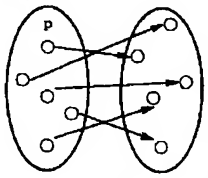
→ FindLoop(FindTrail(FindTrail(pi, (True))∧(p∧¬q), ¬q)∧¬q)

→ FindLoop(FindTrail(FindTrail(pi, (True)∧p, ¬q)∧¬q)

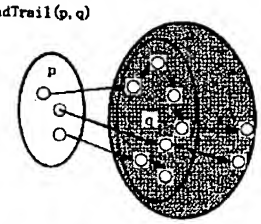
【図７】

基本手続きを説明する図

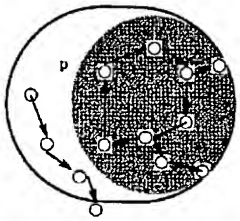
(a) FindTrans(p)



(b) FindTrail(p, q)

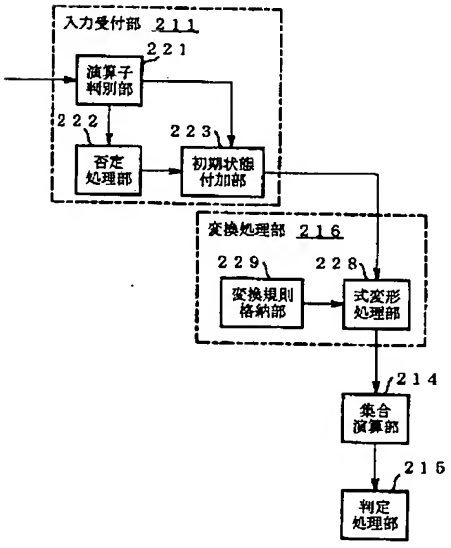


(c) FindLoop(p)



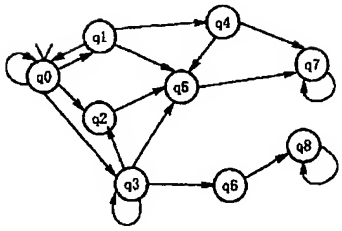
【図８】

請求項７のプロパティ検証装置の実施形態を示す図



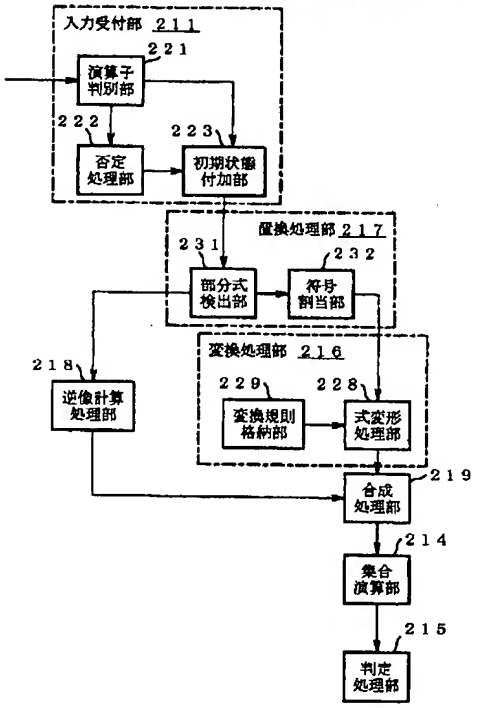
【図１１】

有限状態機械の例を示す図



【図９】

請求項８のプロパティ検証装置の実施形態を示す図



【図１２】

検証手続きの処理時間を説明する図

モデル	関数化なし		関数化あり	
	像計算	逆像計算	像計算	逆像計算
atm-sw	8.9	21.8	10.9	202.5
dh-1	0.0	0.8	0.3	1.3
dh-2	13.8	75.4	46.8	>10000
vpp	space	space	3.0	4135.1
pipe-s	10.7	14.8	0.6	387.9
pipe-d	space	space	200.4	>20000